

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«До захисту допущено»

Науковий керівник кафедри

_____ І.А. Дичка

«__»_____ 2019 р.

Дипломний проект

на здобуття ступеня бакалавра

з напрямку підготовки 6.050103 «Програмна інженерія»

**на тему: «Автоматизована система динамічного визначення змін у
стані об'єкта. Комунікативний модуль»**

Виконав:

студент IV курсу, групи КП-51

Охримчук Денис Дмитрович

Керівник:

Ст. викладач кафедри ПЗКС, к.т.н.,

Люшенко Л.А.

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н.,

Онай М.В.

Рецензент:

Доцент кафедри ММСА ПСА, к.ф.-м.н.,

Шубенкова І.А.

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____

Київ – 2019 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки (програма професійного спрямування) –
6.050103 «Програмна інженерія»

ЗАТВЕРДЖУЮ

Науковий керівник кафедри

_____ І.А. Дичка

«__» _____ 2018 р.

ЗАВДАННЯ

на дипломний проект студенту

Охримчуку Денису Дмитровичу

1. Тема проекту «Автоматизована система динамічного визначення змін у стані об'єкта. Комунікативний модуль», керівник проекту Люшенко Леся Анатоліївна, к.т.н., старший викладач, затверджені наказом по університету від «22» травня 2019 р. №1331-С
2. Термін подання студентом проекту «19» червня 2019 р.
3. Вихідні дані до проекту: див. Технічне завдання.
4. Зміст пояснювальної записки:
 - аналіз існуючих рішень поставленої задачі;
 - обґрунтування вибору засобів реалізації;
 - розроблення алгоритмів та модулів;
 - аналіз та тестування розробленої системи.
5. Перелік обов'язкового графічного матеріалу:
 - діаграма варіантів використання (креслення);
 - ER діаграма (креслення);
 - інтерфейс користувача (плакат);
 - зовнішня архітектура системи (плакат).

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М.В., доцент		

7. Дата видачі завдання «31» жовтня 2018 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів проекту	Примітка
1.	Вивчення літератури за тематикою проекту	01.10.2018	
2.	Розроблення та узгодження технічного завдання	30.11.2018	
3.	Аналіз аналогів системи та розробка вимог	14.01.2019	
4.	Проектування зовнішньої архітектури системи	01.02.2019	
6.	Проектування внутрішньої архітектури, обрання засобів реалізації	15.02.2019	
8.	Розробка бази даних. Створення структури бази даних	15.03.2019	
9.	Завершення роботи над базою даних. Розробка інтерфейсу	22.03.2019	
10.	Завершення роботи над інтерфейсом. Тестування розробленого модулю	1.04.2019	
11.	Тестування системи, аналіз її роботи	20.04.2019	
12.	Підготовка матеріалів першого розділу дипломного проекту	01.05.2019	
13.	Підготовка матеріалів другого розділу дипломного проекту	08.05.2019	
14.	Підготовка матеріалів третього розділу дипломного проекту	15.05.2019	
15.	Підготовка матеріалів четвертого розділу дипломного проекту	20.05.2019	
16.	Оформлення документації дипломного проекту	24.05.2019	

Студент

Д.Д. Охримчук

Керівник проекту

Л.А. Люшенко

АНОТАЦІЯ

Дана робота присвячена розробленню комунікативного модулю для системи динамічного визначення змін у стані об'єкту. Розроблена система орієнтується на аналіз змін у стані мішеней для підвищення ефективності та швидкості тренувань стрільців.

У роботі був проведений аналіз систем, що вирішують проблеми стрільців та допомагають їм у тренуваннях. Були відмічені їх переваги та недоліки, функції що схожі системи надають користувачу.

Були сформовані вимоги до системи, що розробляється, із урахуванням проведеного аналізу існуючих рішень. Розроблено структуру бази даних, сформована логіка для взаємодії бази даних із іншими модулями системи. Розглянута та реалізована ідея побудови динамічного інтерфейсу, описана логіка взаємодії інтерфейсу із іншими модулями системи. Під час роботи над системою розробниками була підтримана ідея універсальності та гнучкості системи, тобто проектування та реалізація системи проводилися з урахуванням того, що зміни і покращення у роботі системі повинні проводитися без обов'язкових змін у структурі системи.

У результаті роботи над дипломним проектом розроблено систему, що дозволяє отримувати інформацію щодо змін у стані мішеней. Комунікативний модуль складається із двох частин – інтерфейсу та бази даних. Інтерфейс даної системи дозволяє користувачу у повній мірі ознайомитись із роботою системи, база даних надає можливість взаємодіяти між собою окремим модулям системи та масштабувати її за необхідністю.

ABSTRACT

This work is devoted to the development of a communicative module for the system of dynamic determination of changes in the state of the object. The developed system focuses on the analysis of changes in the state of the targets for increasing the efficiency and speed of training shooters.

The systems that solve the problems of the shooters and help them in the training have been analyzed. The advantages and disadvantages of these systems have been noted. The functions that the systems provide to the users have been reviewed.

The requirements for the developed system were formed, taking into account the analysis of existing solutions. The database structure is developed, the logic for interaction of the database with other modules of the system is formed. It is considered and realized the idea of constructing a dynamic interface, the logic of interaction of the interface with other modules of the system is described. During the work on the system the developers supported the idea of universality and flexibility of the system, so design and implementation of the system were conducted taking into account that changes and improvements in the work of the system should be carried out without compulsory changes in the structure of the system.

As a result of work on the diploma project the system that allows you to receive information on changes in the state of the targets has been developed. Developed module consists of interface and database. The interface of this system allows the user to interact with the work of the system, the database provides the ability to interact with each module of the system and scale it as needed.

АННОТАЦИЯ

Данная работа посвящена разработке коммуникативного модуля для системы динамического определения изменений в состоянии объекта. Разработанная система ориентируется на анализ изменений в состоянии мишеней для повышения эффективности и скорости тренировок стрелков.

В работе был проведен анализ систем, решающих проблемы стрелцов и помогают им в тренировках. Были отмечены их преимущества и недостатки, функции что подобные системы предоставляют пользователю.

Были сформированы требования к разрабатываемой, с учетом проведенного анализа существующих решений. Разработана структура базы данных, сформированная логика для взаимодействия базы данных с другими модулями системы. Рассмотрена и реализована идея построения динамического интерфейса, описана логика взаимодействия интерфейса с другими модулями системы. Во время работы над системой разработчиками была поддержана идея универсальности и гибкости системы, то есть проектирование и реализация системы проводились с учетом того, что изменения и улучшения в работе системе должны проводиться без обязательных изменений в структуре системы.

В результате работы над дипломным проектом разработана система, позволяющая получать информацию об изменениях в состоянии мишеней. Коммуникативный модуль состоит из двух частей – интерфейса и базы данных. Интерфейс данной системы позволяет пользователю в полной мере ознакомиться с работой системы, база данных предоставляет возможность взаимодействовать между собой отдельным модулям системы и масштабировать ее по мере необходимости.

ДП.045200-01-90 Автоматизована система динамічного визначення змін у стані об'єкта. Комунікативний модуль. Відомість проекту

Позначення	Найменування	Кіл-ть	Примітка
	Документація проекту		
ДП.045200-02-91	Автоматизована система	4	
	динамічного визначення		
	змін у стані об'єкта.		
	Комунікативний модуль.		
	Технічне завдання		
ДП.045200-03-81	Автоматизована система	62	
	динамічного визначення		
	змін у стані об'єкта.		
	Комунікативний модуль.		
	Пояснювальна записка		
ДП.045200-04-51	Автоматизована система	4	
	динамічного визначення		
	змін у стані об'єкта.		
	Комунікативний модуль.		
	Програма та методика		
	тестування		
ДП.045200-05-34	Автоматизована система	6	
	динамічного визначення		
	змін у стані об'єкта.		
	Комунікативний модуль.		
	Керівництво користувача		
ДП.045200-06-99	Автоматизована система	1	
	динамічного визначення		
	змін у стані об'єкта.		
	Комунікативний модуль.		
	Схема бази даних		

[illegible]

--	--	--	--

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Науковий керівник кафедри

_____ І.А. Дичка

“ ____ ” _____ 2018 р.

**АВТОМАТИЗОВАНА СИСТЕМА ДИНАМІЧНОГО ВИЗНАЧЕННЯ
ЗМІН У СТАНІ ОБ'ЄКТА. КОМУНІКАТИВНИЙ МОДУЛЬ**

Технічне завдання

ДП.045200-02-91

“ПОГОДЖЕНО”

Керівник проекту:

_____ Л.А. Люшенко

Нормоконтроль:

_____ М.В. Онай

Виконавець:

_____ Д.Д. Охримчук

ЗМІСТ

1. Найменування та галузь застосування.....	3
2. Підстава для розроблення.....	3
3. Призначення розробки.....	3
4. Вимоги до програмного продукту.....	3
5. Вимоги до проектної документації.....	4
6. Етапи проектування.....	4
7. Порядок тестування розробки.....	4

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ

Назва розробки: Автоматизована система динамічного визначення змін у стані об'єкта. Комунікативний модуль.

Галузь застосування: інформаційні технології.

2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ

Підставою для розроблення є завдання на дипломне проектування, затверджене кафедрою програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Ігоря Сікорського).

3. ПРИЗНАЧЕННЯ РОЗРОБКИ

Розробка призначена для використання в якості програмно-апаратного комплексу з метою зменшення витрат ресурсів та часу на збір даних з мішені та їх подальший аналіз під час стрілецьких тренувань. Комунікативний модуль системи дозволяє взаємодіяти різними модулями між собою, взаємодіяти із системою через інтерфейс.

4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ

Система має відповідати наступним вимогам:

1. Можливість вести спостереження із власного пристрою.
2. Ідентифікація та аналіз влучень у мішень.
3. Гнучка структура системи.
4. Архітектурна підтримка будь-якої стрілецької зброї.
5. Використання системи одночасно групою користувачів.

5. ВИМОГИ ДО ПРОЕКТНОЇ ДОКУМЕНТАЦІЇ

У процесі виконання проекту повинна бути розроблена наступна документація:

- 1) пояснювальна записка;
- 2) програма та методика тестування;
- 3) керівництво користувача;
- 4) креслення:
 - «Автоматизована система динамічного визначення змін у стані об'єкта. Комунікативний модуль. Варіанти використання»;
 - «Автоматизована система динамічного визначення змін у стані об'єкта. Комунікативний модуль. Схема бази даних».

6. ЕТАПИ ПРОЕКТУВАННЯ

Вивчення літератури за тематикою проекту.....	01.10.2018
Розроблення та узгодження технічного завдання.....	30.11.2018
Аналіз аналогів системи та розробка вимог.....	14.01.2019
Проектування зовнішньої архітектури системи.....	01.02.2019
Обрання засобів реалізації.....	15.02.2019
Проектування внутрішньої архітектури.....	24.02.2019
Розробка бази даних.....	15.03.2019
Розробка інтерфейсу.....	22.03.2019
Тестування розробленого модулю.....	01.04.2019
Тестування системи, аналіз її роботи.....	20.04.2019
Оформлення документації дипломного проекту.....	24.05.2019

7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Науковий керівник кафедри

_____ І.А. Дичка

“ ____ ” _____ 2019 р.

**АВТОМАТИЗОВАНА СИСТЕМА ДИНАМІЧНОГО ВИЗНАЧЕННЯ
ЗМІН У СТАНІ ОБ'ЄКТА. КОМУНІКАТИВНИЙ МОДУЛЬ**

Пояснювальна записка

ДП.045200-03-81

“ПОГОДЖЕНО”

Керівник проекту:

_____ Л.А. Люшенко

Нормоконтроль:

_____ М.В. Онай

Виконавець:

_____ Д.Д. Охримчук

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	4
ВСТУП.....	7
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ	8
1.1. Постановка задачі	8
1.2. Огляд існуючих систем, схожих за призначенням	9
1.3. Аналіз існуючих систем. Розроблення вимог	16
1.4. Висновок.....	19
2. ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ.....	21
2.1. Обґрунтування вибору системи керування базами даних	21
2.2. Обґрунтування вибору засобу реалізації інтерфейсу	28
2.3. Висновки.....	32
3. РОЗРОБЛЕННЯ КОМУНІКАТИВНОГО МОДУЛЮ	33
3.1. Загальний опис системи.....	33
3.2. Архітектура системи.....	35
3.3. Варіанти використання	37
3.4. Особливості системи. Побудова динамічного інтерфейсу	40
3.5. Опис розробленої системи. База даних.....	41
3.6. Опис розробленої системи. Інтерфейс.....	44
3.7. Висновки.....	46
4. АНАЛІЗ ТА ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ	48
4.1. Загальний аналіз розробленої системи	48
4.2. Тестування розробленої системи.....	48
4.3. Порівняння із існуючими продуктами.....	52

4.4. Шляхи вдосконалення та змін системи	53
4.5. Зміна напрямлення роботи системи	55
4.6. Висновки.....	56
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ	59
ДОДАТКИ.....	62

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

СКБД – система керування базами даних.

Стрілець – людина, що робить постріли у ціль.

Мішень – це найбільш поширений варіант цілі у стрільбах. Найчастіше являє собою лист паперу із позначеннями на ньому.

Інсталяція – встановлення програмного забезпечення.

Комп'ютерний зір – теорія і технологія створення машин, які можуть виробляти виявлення, стеження та класифікацію об'єктів.

Балістика кулі – це опис, прогнозування руху кулі у залежності від відстані.

Системи керування базами даних – це системи, що призначені для зберігання та маніпулювання даними.

Реляційна модель – це модель, на основі якої і були побудовані реляційні бази даних. Вона полягає у тому, що збереження даних відбувається у структурованому вигляді. Дані у базі даних розташовані із певним набором відношень, що забезпечують цілісність та доступ до даних.

Нормальні форми – це правила, яких правильна структура реляційної бази даних повинна дотримуватися. Існує шість нормальних форм а також декілька проміжних форм.

NoSQL БД – такі бази даних, що використовують неструктурований підхід до збереження даних. Завдяки цьому такі системи можуть уникнути проблем реляційної моделі побудови баз даних, а саме – проблеми з масштабованістю та продуктивністю за будь яких розмірів. Одною з головних переваг NoSQL над реляційними базами даних є легка горизонтальна масштабованість – тобто поєднання баз за допомогою зв'язку між машинами.

Інтерфейс користувача – це частина системи, що надає можливості взаємодії користувача з системою та обмінюватися даними із користувачем.

Текстові інтерфейси – це інтерфейси, що дозволяють взаємодіяти із системою за допомогою текстових команд та текстових меню. Прикладом такого інтерфейсу є звичайна консоль комп'ютера.

Графічні інтерфейси – це інтерфейси, що дозволяють взаємодіяти із системою за допомогою графічних елементів.

Веб-інтерфейс – це інтерфейс, що будується на основі однієї або декілька веб-сторінок і надає користувачу можливості для взаємодії із системою.

Віконний інтерфейс – це інтерфейс, що будується на основі віконного застосунку.

Архітектура системи – це сукупність зв'язків між частинами системи. В стандарті ISO/IEC/IEEE 42010:2011 “Systems and software engineering – Architecture description” архітектура – це властивості системи, що втілені в її елементах, зв'язках між ними і принципах проектування [1].

Варіант використання – це опис можливих послідовностей взаємодій між системою та її користувачами, пов'язаних з певною метою [2].

Схема бази даних – це набір об'єктів бази даних, у більшості випадків – таблиць та їх структури [3].

Мануальне тестування – процес тестування, у якому не використовуються ніякі сторонні програмні доповнення. Людина, що проводить тестування системи виконує його власноруч шляхом модулювання імовірних дій користувача системи.

Авторизація – вхід у систему за унікальним логіном та паролем. Пароль ставиться логіну у відповідність.

Скрипт – послідовність дій, що написана розробником для виконання середовищем, у якому запускається. Найчастіше являє собою окремий файл, який використовує програма для запуску відповідної функції.

Скриптова мова програмування – мова програмування, що орієнтується на написання скриптів що потім використовуються у роботі.

Хешування – отримання послідовності біт, що відповідає інформації, яка надається.

API – application programming interface, це опис інструментів для взаємодії із програмою.

Роутер – мережевий пристрій, що керує процесом маршрутизації усередині локальної мережі.

ВСТУП

Багато технологій у світі зараз знаходяться на стадії активного розвитку та дослідження. Однією з технологій, що набуває своєї популярності є технології аналізу станів об'єкту.

Застосування технології динамічного визначення змін у стані об'єкту може бути дуже варіативним. Її можна застосовувати як у дослідженнях, так і в комерційних продуктах. Вона дозволяє автоматично відстежувати зміни, які є суттєвими для задачі у рамках системи, що розглядається.

Дана технологія дозволяє спростити роботу людей, запобігти зайвим витратам часу та підвищити безпеку роботи із деякими об'єктами. Також технології, що автоматизують окремі частини сучасних систем дозволяють запобігти помилок, спричинених людським фактором.

Технологія дозволяє не тільки відстежувати зміни у стані об'єкту, а при відповідних налаштуваннях й сортувати, аналізувати, структурувати та відбирати ті зміни, що цікавлять відповідного користувача.

У даній дипломній роботі розглядається створення такої системи, що динамічно визначає зміни у стані мішеней, аналізує їх та надсилає користувачу у зручній для людини формі.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ

1.1. Постановка задачі

Задача системи динамічного визначення змін у стані об'єкта полягає у знаходженні, встановленні та аналізу різниці у стані об'єкта у різні проміжки часу.

Автоматизована система, яка розробляється в дипломному проєкті, орієнтована на спрощення, підвищення ефективності, загальне покращення процесу тренування на полігонах та інших місцях, призначених для проведення стрільб. Задача системи у рамках тренувань полягає у позиціюванні та аналізі змін у стані мішеней під час тренувань. Система дозволить перевіряти, куди саме влучив стрілець та зберігати ці данні.

Процес використання, а саме: перегляд, контроль, збереження даних, отриманих із системи динамічного визначення змін у об'єкті може бути хаотичним, неструктурованим та неефективним за відсутності вдало спроектованого інтерфейсу та пов'язаної із системою бази даних.

У цій роботі розробляється комунікативний модуль для системи динамічного визначення змін у стані об'єкту. Вирішуються наступні проблеми:

- Неструктуроване збереження даних системи. Буде створено та узгоджено інтерфейс для взаємодії системи позиціювання із базою даних. Також проєктування бази даних буде відбуватись із урахуванням використання системи у різних ситуаціях.
- Низька ефективність використання системи. Буде закладений простий спосіб оновлення інтерфейсу із урахуванням частих змін у системі. Для покращення структуризації інформації, буде відбуватись попередня обробка інформації перед її отриманням. Проєктування інтерфейсу буде виконуватись з урахуванням перегляду різної за наповненням інформації із бази даних.

- Важкий у розумінні контроль системи. У інтерфейс та систему загалом буде вбудовано прості інструкції для користування та розгортки системи. Також інтерфейс буде проектуватись із урахуванням того, що її будуть використовувати як досвідчені люди, так і нові користувачі.

1.2. Огляд існуючих систем, схожих за призначенням

1.2.1. MANTISX SHOOTING PERFORMANCE SYSTEM

MANTISX – це автоматизована система, яка включає: пристрій, що прикріплюється до зброї, та програмного застосунку, що надає інтерфейс та зв'язок із пристроєм. Застосунок орієнтований на такі операційні системи, як IOS та Android. Інтерфейс системи наведено на рис. 1, а пристрій, що приєднується до зброї – на рис. 2 [4].

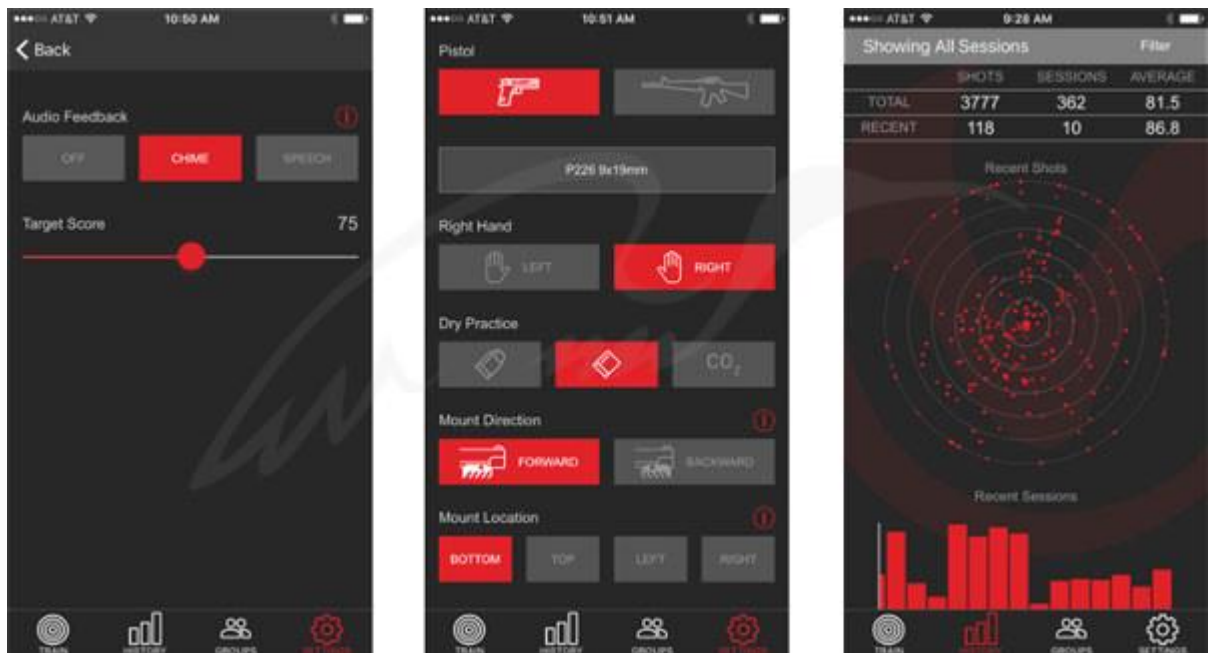


Рис. 1. Інтерфейс системи MANTISX

Для роботи програмно-апаратна система MANTISX потребує:

- приєднати пристрій MANTISX на зброю;

- провести інсталяцію програми MANTISX на смартфон або інший мобільний пристрій, включити дану програму та пристрій, поєднати пристрій із смартфоном.

Далі система пропонує відстежувати ефективність пострілів, ідентифікувати сфери для поліпшення, відстежувати прогрес у часі.



Рис. 2. Пристрій системи MANTISX

Ця програмно-апаратна система використовує певні технології при роботі:

- Технологія зв'язку. Пристрій системи з'єднується з програмною частиною, що знаходиться на мобільному пристрої користувача, за допомогою протоколу Bluetooth (Bluetooth Low Energy або Bluetooth 4.0).
- Збереження даних. Збереження даних системи проходить локально на мобільному пристрої користувача. Для цього застосовується локальна база даних на пристрої, що використовує файлову систему для роботи.
- Інтерфейс. Інтерфейс являє собою мобільний додаток. Для цього були використані технології для побудови мобільних додатків.

Можливості системи MANTISX:

- авторизація за користувачем;

- використання системи у декількох режимах:
 - «сухі» постріли – тобто постріли без застосування боєприпасів;
 - звичайна стрільба;
 - стрільба із зброї, що заправляють CO₂ – тобто пневматичної;
- проводити аналіз руху зброї під час пострілу;
- надавати систему оцінювання, що орієнтується на правильність виконання пострілу;
- проводити аналіз пострілів;
- проводити аналіз покращення ефективності рухів та виконання вправ;
- надавання вправ для тренування;
- надавання порад для покращення техніки стрільби.

Дана система має певні недоліки:

- потрібно встановлювати на зброю сторонній пристрій. Існує зброя, що не має для цього місце, та також це змінює баланс та габарити зброї;
- неможливо установити, куди саме був проведений постріл і перевірити факт влучання;
- для кожної окремої людини потрібен свій пристрій;
- відсутність можливості використання системи на інших платформах, окрім Android та IOS;
- відсутність можливості аналізування покращення влучності стрільця.

1.2.2. SCATT Shooting trainers

SCATT – автоматизована система, що являє собою пристрій, що кріпиться до зброї, та програмного забезпечення, що аналізує постріли. Частина системи наведені на рис. 2 та 3 [5].

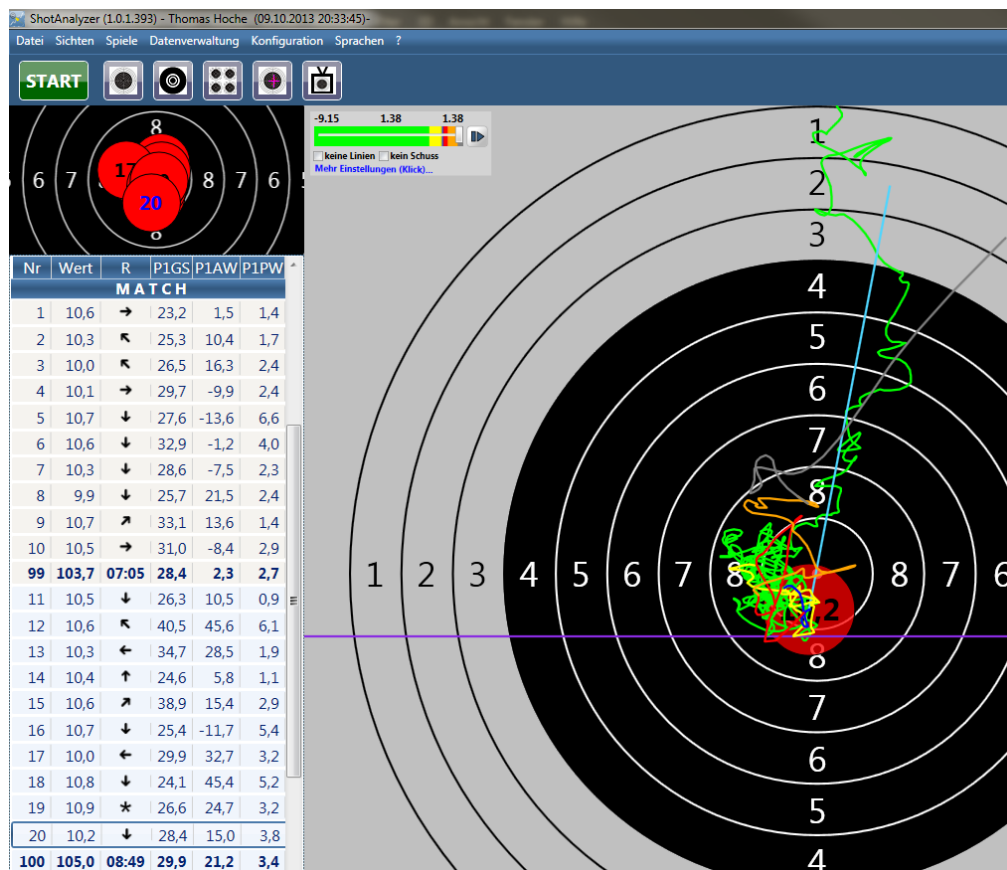


Рис. 3. Интерфейс системы SCATT



Рис. 4. Пристрій системи SCATT

Системи SCATT є декількох видів:

- система SCATT для «сухої» стрільби, тобто без використання боєприпасів;
- система SCATT, що підтримує режими роботи як «сухої» стрільби, так і з використанням боєприпасів.

Система SCATT працює і розроблена із застосуванням таких технологій:

- Технологія зв'язку. Пристрій системи з'єднується з програмною частиною, що знаходиться на комп'ютері користувача за допомогою USB-протоколу, (USB 2.0 та USB 3.0) та за допомогою протоколу Bluetooth.
- Збереження даних. Збереження даних системи проходить локально на комп'ютері користувача. Використовується база даних, що не потребує окремої розгортки, а саме – локальна база даних, що використовує файлову систему для збереження даних системи.
- Інтерфейс. Для побудови інтерфейсу системи було використані технології для побудови віконних доданків.

Система SCATT у обох варіантах використовує технології лазерного вказівника для ідентифікації положення зброї відносно цілі.

Для використання системи необхідно:

- встановити пристрій на зброю;
- встановити програмне забезпечення на комп'ютер (платформи – Windows, MAC OS);
- обрати ціль, у яку будуть проводитись постріли із списку;
- встановити ціль у добре освічене місце;
- відкалібрувати пристрій, встановлений на зброї відносно цілі, що використовується.

Система надає можливості:

- тренувати рух зброї під час пострілу;

- аналізувати помилки при використанні зброї;
- аналізувати влучність пострілів у ціль.

Збереження даних, отриманих під час роботи системи виконується локально на комп'ютері.

Недоліками цієї системи є:

- встановлення на зброю стороннього пристрою;
- відсутність можливості використання декількома людьми водночас;
- робота системи лише з однією цілю;
- довга інсталяція системи для роботи із лише однією цілю;
- відсутність можливості користуватись системою із мобільних пристроїв;
- відсутність підтримки інших ОС окрім Windows та MAC OS;
- обмежена дальність роботи системи.

1.2.3. MARKSMAN Training System

Система MARKSMAN ST-2 – симулятор, спеціально розроблений для тренування стрільців. Це програмно-апаратна система, що складається з проекторів, що моделюють місцевість на площині, та пристроїв, що кріпляться до зброї. Пристрій, що кріпиться до зброї, наведений на рис. 5 [6].

Дана система постачається у різних комплектаціях:

- Classic. Класична комплектація, дозволяє використовувати всі види зброї для тренувань за різними дисциплінами та із виконанням різних вправ. Є компактною і легкою у транспортуванні та розгортці у місцях обмеженого простору.
- Classic wide. Комплектація із усіма перевагами класичної, та із збільшеним кутом зору користувача. Додана можливість стрільби по тарілкам.

- Premium pro. Найкраща комплектація, зі збільшеною картинкою та якістю зображення.

Усі комплектації використовують однакові технології, змінюється якість та збільшуються функціональні можливості системи.

Для роботи система потребує:

- встановлення проектору, що постачається із системою;
- встановлення пристрою на зброю;
- обрати режими роботи.



Рис. 5. Пристрій системи MARKSMAN

Пристрій, що кріпиться на зброю використовує технології комп'ютерного зору, без використання лазера. Пристрій аналізує за допомогою камери місце, у яке був проведений постріл.

Керування системою можливо як за допомогою миші та клавіатури, так і за допомогою голосу (наприклад, запуск тарілок для відстрілу).

Миша та клавіатура може підключатися до проектору, керування проходить через екран, що він проектує. Отже, при побудові інтерфейсу до пристрою були застосовані окремі елементи голосового та графічного інтерфейсу.

Система повинна зберігати локально данні із вправами, зображеннями та іншою необхідною для роботи інформацією. Для цього система використовує базу даних, що розташована локально на пристрої, що проектує зображення та цілі на екран.

Ця система надає можливості:

- аналізувати куди був проведений постріл, як саме летіла потенційна куля, тобто – її балістику;
- тренуватись як на мішенях, так і на рухомих об'єктах.

Перевагами даного рішення є можливості тренувань у приміщеннях, широкі можливості із аналізу балістики пострілів та відпрацювання потенційних ситуацій, що можуть статися на охоті та змаганнях.

Недоліками системи можна виділити те, що вона не передбачає збереження даних та не може працювати із справжніми пострілами та тренуваннями. Також ця система призначена більше для тренувань людей із певним видом зброї, а саме – із рушницями.

1.2.4. Вирішення проблеми без сторонніх пристроїв

Більшість стрільців та стрілецьких кооперацій організують та проводять тренування власноруч, без програмно-апаратних засобів та систем.

Перевагами подібного рішення є можливості побудови тренувань максимально задовольняючи бажанням тренерів та учнів, але відсутність автоматизації під час тренувань зменшує їх ефективність та динамічність.

При тренуваннях без допоміжних систем багато часу на тренуваннях витрачається на:

- збір інформації із мішеней;
- заміну цілей;
- координацію стрілецьких груп для забезпечення безпеки тренувань.

1.3. Аналіз існуючих систем. Розроблення вимог

Розглянуті системи надають можливості спрощення та покращення процесу тренувань стрільців. Вони надають додаткові можливості при тренуваннях, та відстежують інформацію, що може бути цікавою та необхідною при тренуваннях стрільцям та їх тренерам.

Недоліками систем можна назвати стислість у використанні, низьку гнучкість та зосередженість на окремих стрільцях, а не групах. Імплементация даних систем є досить обмеженою, через деяку обмеженість технологій, що використовуються. Також дані системи мають певні обмеження, що є досить критичними при тренуваннях на полігонах великих груп людей.

Із урахуванням існуючих систем та проблем, що розглядаються, сформульовано вимоги до системи та окремих її частин.

1.3.1. Загальні вимоги до системи

Основною ціллю системи є полегшення та підвищення ефективності процесу тренувань. Отже, система буде займатися процесом збору даних з мішені та аналізом цих даних шляхом заміни збору даних вручну на автоматизовану ідентифікацію та аналіз точності влучення, а також збільшення рівня безпеки та ефективності тренувань.

Можна сформулювати такі вимоги:

- Система має надавати змогу користувачеві вести спостереження з певного пристрою (комп'ютера або планшета) за однією або декількома камерами одночасно (кількість камер залежить від технічних можливостей самого пристрою). Кожна з камер системи має вести за однією або декількома мішенями.
- Система має ідентифікувати влучення у мішені, та:
 - ідентифікувати влучення;
 - проводити розрахунки точності влучення;
 - запам'ятовувати послідовність влучень;
 - виділяти на кадрах, що формує система, влучення;
 - групувати влучення у зручний спосіб;
 - виводити інформацію щодо влучень на інтерфейс;
 - надавати доступ до влучень, що були збережені у системі, та їх аналізу.

- Система має бути достатньо гнучкою, щоб змінюватись без повної переробки під окремі вимоги потенційного користувача.
- Використання системи повинне підтримувати застосування різної стрілецької зброї.
- Має бути можливість використання системи одночасно групою людей із різних пристроїв.

1.3.2. Вимоги до бази даних розроблюваної системи

База даних у рамках системи має надавати наступні можливості:

- зберігання даних, отриманих із камер;
- обробка інформації;
- конвертація інформації у вигляд, зручний для перегляду користувачем інтерфейсу;
- зберігання статичної інформації о системі;
- надання даних інтерфейсу.

Для збереження гнучкості та надійності системи, база даних повинна задовольняти наступним нефункціональним вимогам:

- Структура бази даних повинна відповідати щонайменше трьом нормальним формам для побудови вдалої та усунення надлишковості при збереженні [7].
- Зберігання даних із камер повинне бути реалізоване через збережені процедури.
- Обробка та робота із інформацією повинна виконуватись за рахунок окремих модулів, що можливо змінювати за бажанням.
- Доступ інтерфейсу до бази даних повинен виконуватись за рахунок збережених процедур.
- Зміни у системі не повинні приводити к тому, щоб з'являлась необхідність у зміні, реструктуризації бази даних, і навпаки – зміни

в базі даних не повинні приводити к змінам у інтерфейсі та модулях із камерами.

1.3.3. Вимоги до інтерфейсу розроблюваної системи

При роботі із інтерфейсом користувачі повинні мати такі можливості:

- переглядати серії пострілів;
- починати нову серію пострілів;
- переглядати стани мішеней, зареєстрованих у системі;
- оновлювати інтерфейс за бажанням, а отже - й отримувати актуальну інформацію при оновленні;
- переглядати довідки про розгортку системи;
- переглядати довідки про роботу із системою.

На інтерфейс накладені наступні нефункціональні вимоги:

- для вибору серії пострілів використовується окремий елемент інтерфейсу, що надає доступ до списку усіх серій пострілів у системі;
- довідки повинні відкриватись у окремому вікні, так як є статичним елементом інтерфейсу, тож не потребують оновлення;
- перегляд стану мішеней потрібен полягати у перегляді актуального фото мішені;
- при початку нової серії автоматично обирати нову серію із списку у інтерфейсі.

1.4. Висновок

Було проаналізовано існуючі пристрої та програмні системи, задачею яких є покращення ефективності тренувань стрільців. Структуровано можливості що вони надають, та недоліки систем.

Були розглянуті та проаналізовані технології, що використовуються для роботи та розроблення існуючих рішень.

Були сформовані вимоги до системи з урахуванням аналізу проблеми та існуючих програмно-апаратних рішень.

Розроблення системи та обрання засобів реалізації будуть враховувати результати аналізу схожих за призначенням систем та вимоги до системи.

2. ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ

Засоби реалізації бази даних та інтерфейсу необхідно обирати із урахуванням фізичних частин системи, що використовуються для розгортки, та програмного забезпечення, розташованого на пристроях користувачів.

Фізичним сервером, на якому буде розгортатись база даних та інші компоненти, необхідні для взаємодії системи із користувачами, є сервер, підключений до Wi-Fi роутеру. Даний сервер має серверну операційну систему Armbian Linux. Це одна з вільних серверних операційних систем. Є UNIX – подібною. Сервер має обмежені ресурси.

Система повинна надавати можливість користувачам із різних пристроїв із різними ОС мати доступ до системи.

2.1. Обґрунтування вибору системи керування базами даних

Основними цілями використання баз даних є:

- зберігання необхідних даних;
- доступ до даних за запитом;
- забезпечення цілісності даних;
- уникнення дублювання і надмірності даних.

Під впливом часу та різних задач було розроблено багато різних систем керування базами даних (СКБД) та багато програм для роботи із ними [3].

Сучасні СКБД можна поділити на дві групи, а саме:

- реляційні, тобто керуючі базами даних реляційної моделі;
- NoSQL, керуючі базами даних неструктурованої моделі [8].

Реляційні СКБД використовують строгі описи для побудови баз даних, а саме – схеми, що описують таблиці, відношення між ними та їх поля. Таблиця – основна структурна одиниця у реляційних СКБД. Таблиці

у реляційний СКБД використовують модель вертикальних стовпців із унікальними іменами та горизонтальних строк із даними [7, 9].

Із використанням NoSQL баз даних системи можуть уникнути проблем реляційної моделі побудови баз даних, а саме – проблеми з масштабованістю та продуктивністю за будь яких розмірів. Одною з головних переваг NoSQL над реляційними базами даних є легка горизонтальна масштабованість – тобто поєднання баз за допомогою зв'язку між машинами [8].

Для NoSQL притаманні гнучкі схеми даних, а для їх розроблення застосовуються різні моделі, наприклад – графові, ключ-значення та ін. Вибір моделі NoSQL – це дуже важливе рішення, так як кожна з моделей найкраща у окремому колі використання [10].

У роботі розглядається система, що не потребує розгортання розподіленої бази даних. У системі є центральний сервер, на якому розташована база даних та який відповідає за збереження даних та взаємодію модулів системи між собою. У системі було вирішено використовувати реляційну базу даних.

2.1.1. СКБД Oracle Database

База даних Oracle (Oracle DB) є реляційною системою управління базами даних (RDBMS) від корпорації Oracle. Спочатку розроблена Лоуренсом Еллісоном та іншими розробниками. Являє собою один з найбільш надійних і широко використовуваних двигунів реляційних баз даних.

Oracle Database є першою базою даних, розробленою для корпоративних мережесхем обчислень, є гнучким і економічно ефективним способом управління інформацією та додатками. Enterprise grid computing створює великі пули стандартного, модульного зберігання і серверів. Завдяки цій архітектурі кожна нова система може бути швидко забезпечена з пулу компонентів. Немає потреби в пікових робочих навантаженнях, тому

що потужність може бути легко додана або перерозподілена з пулів ресурсів за необхідності [11, 12, 13, 14].

База даних має логічні структури та фізичні структури. Оскільки фізичні та логічні структури є окремими, фізичним зберігання даних можна управляти, не впливаючи на доступ до логічних структур зберігання [13].

Переваги:

- підтримує більшість із існуючих платформ;
- підтримує об'єктно-орієнтований підхід;
- легко масштабується;
- має розвинену систему індексації;
- ефективна система хешування.

Недоліки:

- складна у розгортанні;
- займає багато місця на сервері;
- потребує високу кваліфікацію розробника для вдалого проектування та підтримки із використанням усіх переваг СУБД;
- дорога при комерційному застосуванні [14, 15].

Для розгортання даної системи необхідно мати певну кваліфікацію та бюджет. Також дана система є занадто об'ємною та вимогливою для апаратного забезпечення, на якому буде розгортатися база даних.

Отже, дана СКБД має певні переваги та недоліки. Потужна при вдалому використанні, але є занадто складною та коштовною для використання її у рамках системи, що розробляється.

2.1.2. СКБД MS SQL Server

SQL Server є системою управління реляційними базами даних Microsoft. Це повнофункціональна база даних, що розроблена та використовується як конкурент до Oracle Database (DB) і MySQL.

Як і всі основні СКБД, SQL Server підтримує ANSI SQL, стандартну мову SQL. SQL Server також містить Transact SQL (T-SQL), що є власною

реалізацією мови SQL. SQL Server Management Studio (SSMS) є основним інструментом для роботи із SQL Server. SSMS підтримує 32-бітові і 64-розрядні платформи розроблення [16].

SQL Server постачається в декількох версіях із різним набором функцій та ціною для користувачів із різними потребами:

- Версія «Enterprise». Призначено для великих підприємств, що мають певні складні вимоги до роботи із даними, їх збереження та підтримкою веб технологій. Має всі можливості SQL Server, є найдорожчою версією із доступних.
- Версія «Standard». Призначено для середніх та малих підприємств та організацій, підтримує різні технології зберігання даних та електронної комерції.
- Версія «Workgroup». Призначено для малих підприємств та організацій. Немає обмежень щодо розміру збереження даних та кількості користувачів. Використовується переважно для веб-серверів.
- Версія «Express». Безкоштовна версія MS SQL Server. Має найменшу кількість функцій, а також має обмеження на розмір бази даних та кількості користувачів. Використовується переважно для заміни роботи із базою даних «Access».

Переваги даної СКБД:

- професійне та зручне ПЗ для роботи із базою даних;
- висока надійність та можливості відновлення даних;
- має великий простір для налаштування.

Недоліки:

- є дорогим комерційним продуктом;
- існує розгортка на Linux, але потребує великих затрат ресурсів для підтримки та не всі версії підтримуються;
- SQL Server займає багато ресурсів операційної системи [17].

Дана система є альтернативою Oracle Database, яка задовольняє вимогам великих та малих підприємств щодо можливостей та надійності. Але вимагає грошей для отримання необхідних для використання функціональних можливостей та складна у розгортці на деяких платформах. Також вимагає досить багато ресурсів системи при роботі.

Отже, дана система є дорогою у використанні як у сенсі грошей, так і ресурсів системи. Також не підтримується деякими версіями Linux, що може бути проблемою при модифікації системи, що розробляється. Недоцільно використовувати дану систему [16].

2.1.3. СКБД PostgreSQL

PostgreSQL – це потужна відкрита система СКБД, що використовує мову SQL та розширює її. PostgreSQL було створено в Каліфорнійському університеті в Берклі у 1986 році і розвивається вже понад 30 років.

PostgreSQL має перевірену архітектуру, високу надійність, широкий набір функцій, цілісність даних, можливості до розширення. Окрім того, має багато доступних доповнень та інноваційних рішень. PostgreSQL підтримує усі основні операційні, з 2001 року сумісний з ACID. Має потужні додатки, прикладом до яких можна навести геопросторовий розширювач бази даних PostGIS [18].

PostgreSQL поставляється з широким колом функціональних можливостей, що дозволяє розробникам створювати додатки до системи, адмініструвати та налаштовувати захист цілісності даних, створювати стійкі до відмов та надійні середовища для збереження даних. У даній системі просто керувати даними незалежно від їх обсягу. Також перевагою даної системи є те, що це є системою з відкритим кодом. Це дозволяє створювати власні типи даних та будувати власні функції, писати код з різних мов програмування.

Одною з найголовніших особливостей та переваг даної системи є те, що вона задовольняє стандарту SQL, де це не суперечить традиційним

особливостям СКБД та не призводить до проблемних архітектурних рішень. А отже, більшість функцій, що необхідні даному стандарту, підтримується даною СКБД. Протягом свого існування система розвивається та покращує свою підтримку стандарту SQL. Версія 2018 року задовольняє принаймні 160 із 179 обов'язкових функцій стандарту SQL: 2011 Core. Треба зазначити, що ніяка СКБД ще не задовольняє цьому стандарту повністю.

Отже, перевагами даної СКБД є:

- система дозволяє використання безкоштовно;
- об'єктно-орієнтованість. Можна використовувати потужний інструмент для покращення логіки роботи із базами даних;
- підтримка останніх стандартів SQL синтаксису;
- велика кількість додатків до системи;
- підтримка різних платформ розгортання.

Недоліки:

- складність у налаштуванні зручної роботи із СКБД;
- може працювати із простими операціями повільніше за інші вільні СКБД.

Отже ця система є досить універсальною та легкою у використанні. Має широкі можливості та є легкою у використанні.

Є досить оптимальним вибором для роботи з базами даних у рамках системи, що розробляється [18].

2.1.4. СКБД MySQL

MySQL – це найпопулярніша система управління базами даних SQL що розробляється, розповсюджується і підтримується корпорацією Oracle.

Сервер баз даних MySQL може бути охарактеризований як швидкий, надійний, масштабований і простий у використанні. При повному використанні можливостей MySQL, можна налаштувати параметри, щоб скористатися всіма можливостями пам'яті, потужності процесору. MySQL також може масштабуватися до кластерів машин, об'єднаних у мережу [19].

MySQL Server був розроблений для роботи із великими базами даних швидше ніж інші рішення. Був успішно протестований та використаний у вимогливих виробничих середовищах протягом декількох років.

Дана СКБД постійно розвивається, і пропонує широке коло можливостей. Характеризується безперебійним зв'язком, високою швидкістю та безпекою. Є підходящим рішенням для доступу до баз даних в Інтернеті та по локальній мережі.

Програмним забезпеченням бази даних MySQL є клієнт-серверна система. Вона складається із многопоточного SQL-сервера, що підтримує деякі клієнтські програми, бібліотеки, інструменти адміністрування та широкий спектр інтерфейсів для прикладного програмування (API) [20, 21].

Переваги:

- легкість розгортання;
- простота використання;
- система безкоштовна для використання;
- велика ефективність та швидкодія, у тому числі і з великими обсягами даних.

Недоліки:

- підтримка стандартів SQL неповна;
- деякі версії мають стислі можливості.

Дана система є легкою у використанні, достатньо універсальною та надає достатньо широкі можливості розроблення. Є гарним вибором для використання при розробленні бази даних для системи динамічного позиціонування.

2.1.5. СКБД SQLite

SQLite – це бібліотека, що реалізує автономний, без-серверний, механізм транзакційних баз даних SQL. Код SQLite є загальнодоступним і, отже, вільним у застосуванні до різних цілей – як комерційних, так і

приватних. SQLite є одною з найбільш широко використовуваних баз даних у світі із великою кількістю додатків, включаючи деякі великі проекти.

SQLite являє собою вбудований движок бази даних SQL, і, на відміну від найбільш поширених реляційних баз даних, не має окремого серверного процесу. SQLite використовує файлову систему для роботи і використовує файли для збереження даних. Повна база даних, із даними та таблицями, зберігається у одному файлі. Формат даних файлів є кросплатформним – можна вільно копіювати базу даних між різними системами і навіть із різною розрядністю. Ці особливості роблять SQLite популярним вибором для використання у різноманітних додатках.

SQLite є досить компактною. Навіть якщо всі функції включені, розмір може бути менший за 600 КБ, хоча це залежить від цільових платформ і зроблених налаштувань. Також вона дозволяє налаштовувати швидкість бази – вона працює тим швидше, чим більше пам'яті їй надати.

Також SQLite характеризується великою надійністю і швидким виправленням можливих помилок. Ця система є відкритою і постійно розвивається.

Формат цієї бази даних накладає певні обмеження на використання та на функції системи. Дана система не підтримує багато можливостей, що підтримує звичайна серверна СКБД [22].

SQLite є досить зручною у використанні та надає певні можливості, але через стислі функціональні можливості не є доцільною для використання у системі, що має окремий сервер для розгортки бази даних, а отже – і у системі, що розроблюється.

2.2. Обґрунтування вибору засобу реалізації інтерфейсу

2.2.1. Обґрунтування вибору технології для побудови інтерфейсу

При виборі засобу реалізації інтерфейсу у системі, що розглядається у роботі, спочатку потрібно обрати технологію для побудови інтерфейсу.

Систему передбачається використовувати на полігонах та інших місцях для стрільб. Такі місця характеризуються великим рівнем шуму, тому використання звукових частин інтерфейсу не є доцільним. Тому доцільним є розглядання видів та технологій для побудови графічного інтерфейсу.

Текстовий інтерфейс. Це консольний інтерфейс, вся взаємодія із системою проходить за допомогою текстових повідомлень в обидві сторони. Не є доцільним для використання у системі, що розглядається, бо є незручним у використанні та складно сформувати подібний інтерфейс у кросплатформному варіанті.

Графічний віконний інтерфейс. Являє собою окремий програмний модуль. Є зручним у використанні, взаємодія із системою проходить за допомогою технологій, що закладені усередину даного інтерфейсу. Є вдалим вибором, але складно зробити застосунок, що буде підтримуватись майже усіма операційними системами. Також потрібно розуміти, що на місцях розташування стрілецької системи може не бути зв'язку для завантаження необхідного у роботі застосунку.

Графічний веб інтерфейс являє собою веб-сайт із елементами керування, є зручним у використанні, взаємодія з системою проходить за допомогою технологій, розташованих в середині веб-серверу. Підтримується усіма операційними системами, що мають браузер. При розташуванні даного інтерфейсу на локальному веб-сервері, даний вид інтерфейсу не буде потребувати від користувачів встановлення ніяких сторонніх застосунків для роботи із системою. Є вдалим вибором для використання у системі, що розробляється у рамках дипломної роботи.

2.2.2. Обґрунтування вибору програмного забезпечення для реалізації інтерфейсу

Інтерфейс користувача — це частина системи, що відповідає за взаємодію користувача з системою та обмін даними із користувачем [23].

Правил та принципів для того, щоб створити інтерфейс якомога комфортнішим для користувача дуже багато. Основні правила такі:

1. Інтуїтивно зрозумілий інтерфейс. Інтерфейс повинен бути зрозумілим користувачу.
2. Структурованість. Інтерфейс повинен мати структуру з урахуванням потреб користувача.
3. Зворотній зв'язок. Система повинна повідомляти користувачу про свої помилки, некоректність роботи.
4. Оптимальна інерційність. Система не повинна різко реагувати на помилки користувача та надавати можливість користувачу виправити свої дії.
5. Доступність. Усі елементи та функції, що необхідні користувачу у окремий етап роботи із системою повинні бути доступні та знаходитися в полі зору користувача.
6. Одноманітність стилю. Якщо користувачу у рамках роботи з системою потрібно взаємодіяти з декількома вікнами та елементами інтерфейсу, вони повинні бути виконані у єдиному стилі для спрощення роботи користувача [9, 24].

Інтерфейс користувача буває багатьох видів. Окрім звичайного візуального існують також тактильні, голосові, матеріальні, жестові [25].

У програмно-апаратних системах найчастіше використовують саме графічний інтерфейс. Наприклад розглянуті у попередніх розділах системи використовують саме графічний інтерфейс та деякі функції голосового.

Візуальні інтерфейси мають своє розділення. Зазвичай їх поділять на дві групи – текстові та графічні.

Графічні інтерфейси також поділяються на групи, основними є:

- веб-інтерфейс;
- віконний інтерфейс.

Перевагами веб інтерфейсу є легко досяжна кросплатформність даного інтерфейсу, так як майже усі операційні системи мають браузер.

Також не потрібно встановлювати на комп'ютер додаткові програми. Недоліком даного інтерфейсу є безпека – потрібно приділяти увагу тому, що люди можуть дивитись внутрішню структуру веб-сторінок.

Віконний інтерфейс є привабливим та простим способом для побудови інтерфейсу, так як існує багато застосунків та платформ, що дозволяють їх розроблювати. Наприклад Visual Studio, Eclipse та ін. Перевагами даного способу є простота, гнучкість розроблення, безпека, що досягається тим, що внутрішня структура програми та код захований усередині цього інтерфейсу без можливості доступу. Недоліками є те, що важко розробити кросплатформний інтерфейс подібного плану, а саме щоб вірно працював як на мобільних пристроях, так і на комп'ютерах.

Стандартними засобами реалізації веб-інтерфейсу є PHP, CSS, HTML, Java Script.

PHP – скриптова мова, що дозволяє на боці веб-серверу, на якому розташований сайт, виконувати обчислення, підтримувати зв'язок із базами даних та інше.

CSS – мова розмітки, що дозволяє інтерфейсу виглядати сучасно, пропорційно. Також дозволяє розташовувати елементи інтерфейсу у тому порядку, якому треба.

HTML – мова розмітки, що будує елементи інтерфейсу.

Java Script – скриптова мова, що дозволяє виконувати обчислювання та інші дії із інтерфейсом на пристрої користувача,

При обранні засобів реалізації веб-інтерфейсу користувача потрібно уникати частин, що будуть виконуватися на боці пристрою користувача. Це впливає з того, що система орієнтується на широке коло користувачів, із різними типами та моделями пристроїв.

Тобто використання Java Script не є доцільним у даній системі, так як пристрої можуть не підтримувати дану мову на своєму боці.

2.3. Висновки

Були розглянуті та проаналізовані варіанти технологій щодо можливості використання під час розроблення бази даних та інтерфейсу для системи динамічного визначення змін у стані об'єкту.

СКБД MySQL та PostgreSQL задовольняють усім вимогам системи, що розроблюється і є вдалим вибором.

Обрано MySQL як систему керування базами даних тому, що вона не потребує багато ресурсів при використанні, є простою у використанні та інсталяції, надає достатні можливості для побудови бази даних, що потрібна у рамках системи.

Обрано веб-інтерфейс як технологію побудови інтерфейсу до системи. Даний вид інтерфейсу дозволяє легко надати кросплатформність у використанні інтерфейсі. Також є достатньо зручним у використанні, та не потребує встановлення окремих застосунків для взаємодії із системою, що є дуже важливим в умовах відсутності зв'язку з інтернетом.

Для створення інтерфейсу було обрано скриптову мову PHP та мови розмітки HTML та CSS.

3. РОЗРОБЛЕННЯ КОМУНІКАТИВНОГО МОДУЛЮ

3.1. Загальний опис системи

Після аналізу систем розроблення було вирішено, розробити систему динамічного визначення змін у стані об'єктів (мішені).

Вона надає можливості аналізу та позиціонування влучень стрільців у мішені. Користувачі можуть побачити як актуальне фото мішені, так і аналіз влучень, що з'явилися під час роботи системи. Приклад того, як виглядає фото мішені у системі, можна побачити на рис. 6. Таблиця із влученнями, що надає система, наведена на рис. 7.

У інтерфейсі користувачі можуть обрати номер серії пострілів, яка їх цікавить, та мішень для перегляду. Також користувачі можуть ознайомитись із довідкою, у якій є інформація про розгортку системи, її можливості та інструкції до користування системою. Інтерфейс також дозволяє починати нову серію пострілів, що дозволяє групувати постріли так, як потрібно користувачеві. Елементи керування системою та інтерфейсом наведені на рис. 8 та 9.



Рис. 6. Актуальне фото мішені із влученням

Your hits			
Id	Target	Time	How far from center (mm)
7763	0	2019-05-11 15:40:25	219.02
7764	0	2019-05-11 15:40:25	197.4
7765	0	2019-05-11 15:40:25	152.31
7766	0	2019-05-11 15:40:25	138

Рис. 7. Таблиця з аналізом влучень у мішень

Series number:

target's number for show:

☒ check current seria automatically

Рис. 8. Панель управління



Рис. 9. Виклик довідки

Для роботи системи необхідно встановити пристрої із камерами, що приєднані до системи, навпроти мішені та направити їх на неї. Це дозволить системі аналізувати стан мішені та результати пострілів. Після підключення головного серверу до локальної мережі, у цій мережі буде можливий доступ

до інтерфейсу. Подальша робота системи не потребує втручання користувача, окрім роботи із інтерфейсом.

3.2. Архітектура системи

3.2.1. Зовнішня архітектура системи

Зовнішню архітектуру розробленої системи можна побачити на діаграмі з рис. 10.

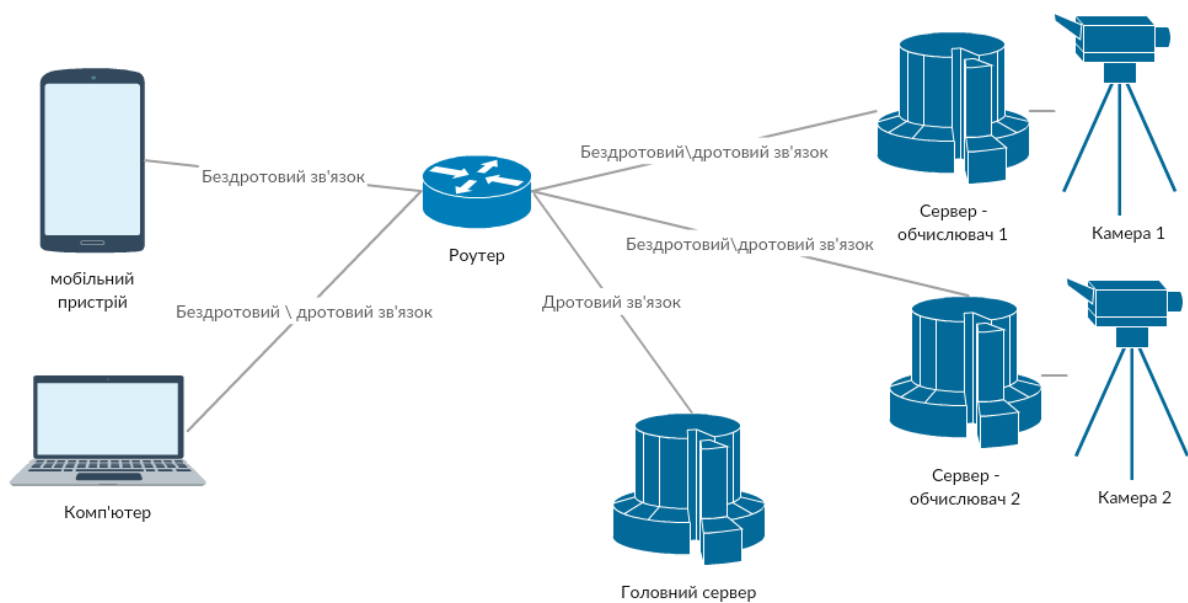


Рис. 10. Зовнішня архітектура системи

Розроблена система складається із таких об'єктів:

- пристроїв користувачів, підключених до мережі;
- роутеру;
- головного серверу;
- серверів-обчислювачів;
- камер.

Камери – це пристрої, що зчитують зображення мішені у режимі реального часу. Вони підключені безпосередньо до серверів-обчислювачів, що мають доступ до зображення з камер.

Сервери-обчислювачі – це об’єкти системи, що підключені до камер. До однієї камери підключений один сервер-обчислювач. Кожен такий сервер отримує зображення безпосередньо з підключених до нього камер та обробляє його у режимі реального часу. Оброблену інформацію за необхідністю по мережі передає до головного серверу. Підключення до головного серверу відбувається по мережевому протоколу SSH.

Роутер – це мережевий пристрій, що дозволяє організувати та налаштувати локальну мережу, у якій роблять та взаємодіють окремі елементи мережі.

Головний сервер – об’єкт системи, що виконує подальшу обробку даних, що отримані із обчислювальних серверів. Він конвертує дані у формат, необхідний користувачу для перегляду та відповідає за налаштування загального вигляду інтерфейсу користувача. Виконує аналіз даних, отриманих під час роботи системи.

Пристрої користувачів – це будь-який пристрій, що підключений до мережі та не є зареєстрованим елементом системи. Пристрої користувачів отримують інтерфейс системи по мережі за допомогою зв’язку із головним сервером по HTTP протоколу.

3.2.2. Внутрішня архітектура системи

Внутрішню архітектуру системи можна побачити на діаграмі з рис. 11.

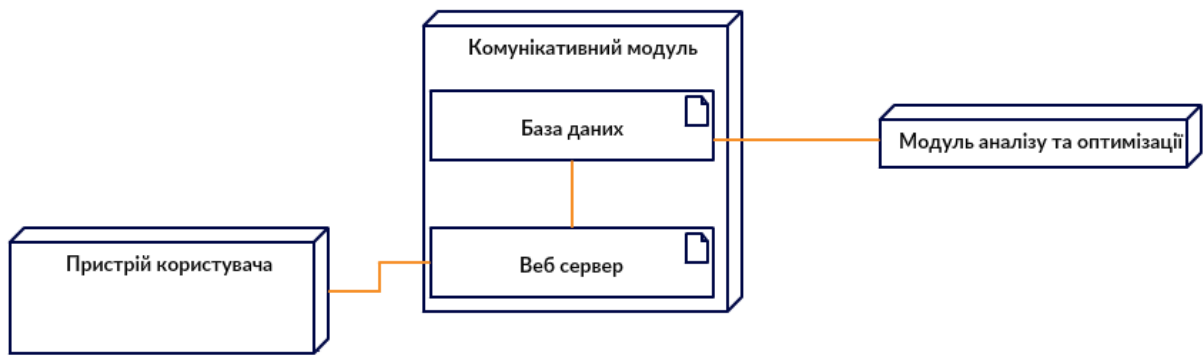


Рис. 11. Внутрішня архітектура системи

Внутрішня архітектура системи складається з двох модулів:

- комунікативний модуль;
- модуль аналізу та оптимізації.

Комуникативний модуль – це центральний модуль, що розташований на головному сервері. Він складається із бази даних та веб-серверу. Веб-сервер застосовує базу даних для передачі інформації користувачеві на пристрій за його запитом.

У ролі модулю аналізу та оптимізації виступає модуль, що розташований на обчислювальному сервері (або серверах). Він відповідає за позиціювання змін у стані мішені та передає дані о змінах, разом із зображеннями мішені, на комуникативний модуль, а саме – у базу даних.

3.3. Варіанти використання

Для пояснення того, як саме повинна та робить система часто використовують варіанти використання. Варіанти використання та їх діаграми дозволяють пояснити, які саме можливості система надає користувачам та як саме вони їм користуються. Діаграма варіантів використання користувача наведена на рис. 12.

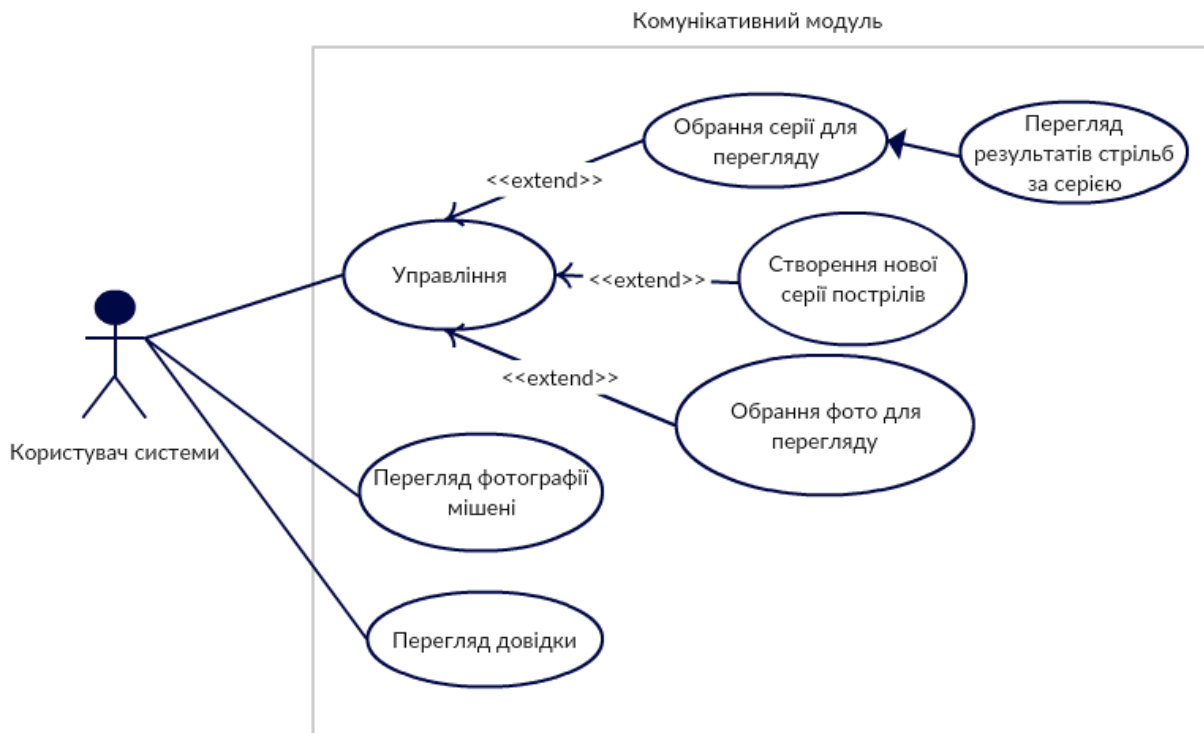


Рис.12. Діаграма варіантів використання користувача системи

На діаграмі використання користувача системи, користувач має у системі такі можливості:

- переглядати фотографії системи;
- переглядати довідку;
- використовувати панель управління.

Переглядання фотографій системи – це варіант використання, що дозволяє ознайомитись із актуальними зображеннями мішеней, за якими ведеться спостереження у рамках роботи системи.

Переглядання довідки – варіант використання, що надає змогу переглядати інструкції до системи.

Використання панелі управління – це варіант використання, що дозволяє налаштовувати користувачеві елементи для отримання у інтерфейсі, а також проводити деякі дії у системі.

У панелі управління користувач може:

- обрати фото для перегляду. Користувач має можливість обрати мішень, яку він хоче спостерігати зараз у системі;

- розпочати нову серію пострілів. У системі створюється нова серія пострілів, і усі подальші влучення у мішені будуть враховуватись саме у цій групі;
- обрання серії пострілів для перегляду. Користувач обирає серію пострілів, за якою виводиться інформація на інтерфейс системи.

Доцільно також розглядати «внутрішню» діаграму використання, так як система складається із двох окремих модулів. Отже, модуль аналізу та оптимізації, що відповідає за позиціювання змін у мішенях має власну діаграму використання із комунікативним модулем. Діаграма зображена на рис. 13.

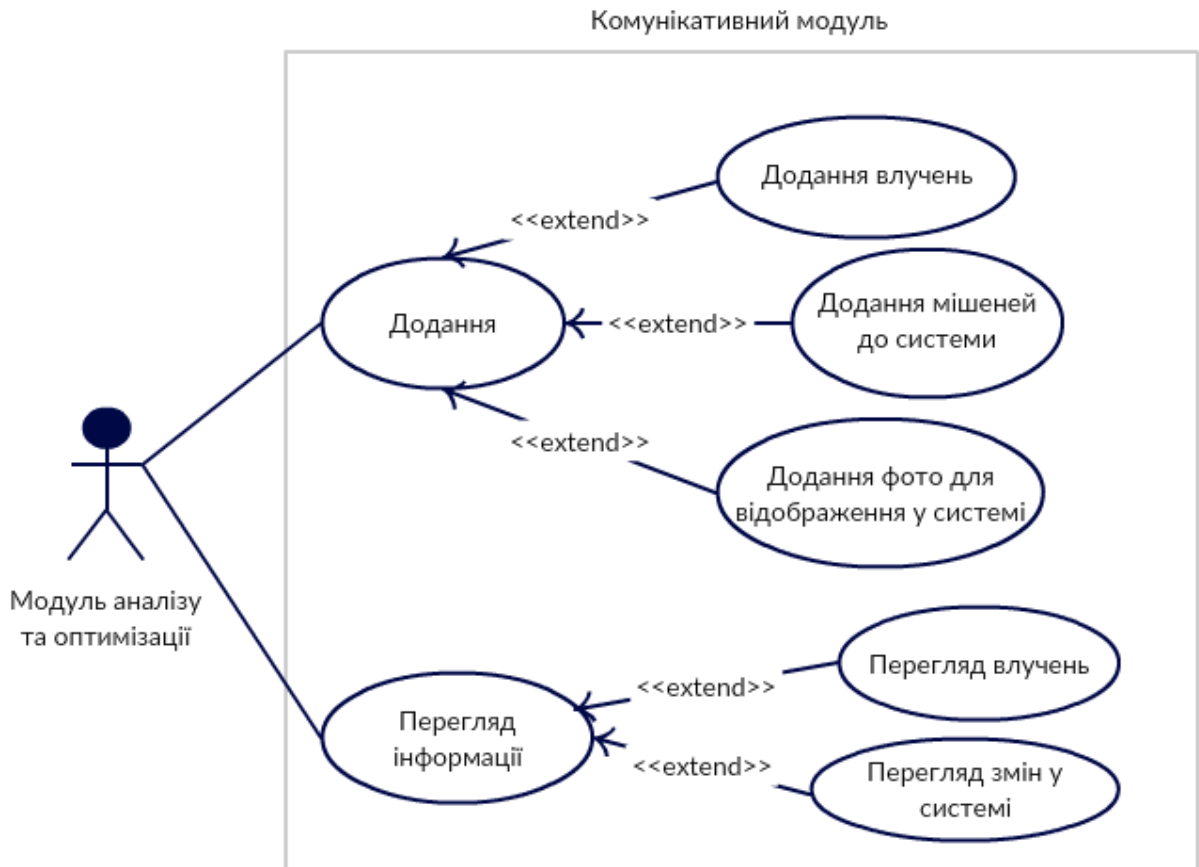


Рис. 13. Діаграма варіантів використання модулю аналізу та оптимізації

Отже, модуль аналізу та оптимізації має змогу додавати данні до системи та переглядати їх:

- модуль аналізу та оптимізації додає дані, які він може отримати із результатів своєї роботи та розгортки системи;
- модуль аналізу та оптимізації переглядає дані, які необхідні йому для синхронізації своєї роботи із системою.

3.4. Особливості системи. Побудова динамічного інтерфейсу

3.4.1. Опис технології побудови динамічного інтерфейсу

Під час розроблення системи були використані нестандартні підходи до створення інтерфейсу та його взаємодії із іншими частинами системи. Була спланована та реалізована ідея побудови динамічного інтерфейсу.

Стандартною реалізацією веб-інтерфейсу є створення HTML файлів на стороні веб-серверу та використання різноманітних скриптових мов для надання функціональних можливостей та гнучкості інтерфейсу.

У розробленій системі на стороні веб-серверу немає HTML коду, він був перенесений до бази даних.

Будування інтерфейсу проходить за рахунок використання запитів до бази даних. База даних у свою чергу з урахуванням параметрів запиту відбирає той HTML код, що задовольняє вимогам, сортує їх та відсилає до веб-серверу. Далі веб-сервер виводить елементи на екран у правильному порядку.

3.4.2. Переваги та недоліки динамічного інтерфейсу

Головними перевагами даного методу є:

- гнучкість системи;
- усі зміни у системі проводяться за допомогою бази даних;
- безпека системи.

Гнучкість системи досягається за рахунок того, що інтерфейс та його наповнення може бути легко змінено лише за рахунок змін частин HTML, що зберігаються у базі даних без змін безпосередньо на сервері та у логіці системи. Отже, для того, щоб система змінила своє призначення із стрілецької у іншу промисловість, достатньо змінити частини HTML. Структура, що була побудована з боку веб-серверу останеться незміною. Також це дозволяє розширювати та доповнювати систему за рахунок додання елементів до бази даних.

Усі зміни у системі проводяться за допомогою бази даних. З цього випливає, що для оновлень та виправлень помилок достатньо оновити базу даних не втручаючись у роботу інших частин системи.

Безпека системи. Система має високий рівень безпеки, так як загальний рівень захищеності баз даних є високим, і вся логіка та інформація про систему зберігається усередині бази даних. Також у зловмисника немає можливості звертатися безпосередньо до файлів інтерфейсу.

Недоліком ідеї та даної реалізації динамічного інтерфейсу є складність у тестуванні такої структури побудови веб-сторінок.

3.5. Опис розробленої системи. База даних

3.5.1. Структура бази даних

Бази даних можуть бути описані за допомогою їх структури. Структура бази даних створюється на етапі її проектування та передбачається, що вона не буде кардинально змінюватися протягом життєвого циклу системи. Структура у свою чергу описується за допомогою схеми, що наведена на рис. 14.

База даних складається із таблиць:

- Таблиця tableParams – місце збереження параметрів, що можуть змінюватись за діями користувача у інтерфейсі.

- Таблиця `browserWindow` – це місце збереження усіх створених частин веб-сторінок. Тобто вона являє собою кеш, що використовується для побудов інтерфейсів за запитами.
- Таблиця `url` – місце збереження усіх запитів із тегами до веб-серверу. Саме за цими запитами формується критерії, із яких будується інтерфейс системи.
- Таблиця `series` – місце збереження усіх серій пострілів, що відбулися під час роботи системи.
- Таблиця `hit` – місце збереження усіх зафіксованих влучень.
- Таблиця `imagesBuffer` – місце збереження зображень мішеней. Задля того, щоб не перевантажувати мережу зайвим трафіком, зберігає усередині лише посилання на зображення – місцезнаходження зображень на сервері.
- Таблиця `targets` – місце збереження інформації об усіх зареєстрованих мішенях у системі.
- Таблиця `partsOfInterface` – місце збереження шаблонів HTML коду, що використовуються при генерації елементів таблиці `browserWindow`.
- Таблиця `logTable` – місце збереження усіх змін, що були проведені із базою даних.

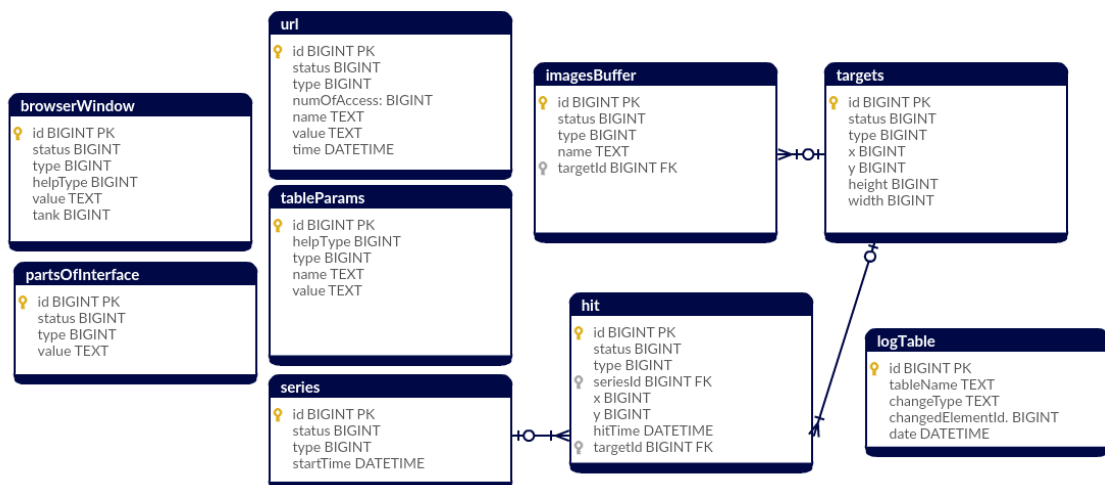


Рис. 14. Схема розробленої бази даних

3.5.2. Взаємодія із базою даних. Особливості будови

Взаємодія із базою даних для усіх частин системи проводиться за допомогою збережених процедур. Перевагами даного підходу є:

- інкапсуляція дій усередині системи;
- надання можливості у легкий спосіб оброблювати та редагувати отриману інформацію;
- незалежність запитів на збереження інформації від структури бази даних;
- підвищена безпека роботи із системою. Окремий користувач бази даних має права не на роботу із таблицями, а на виклик збережених процедур;
- полегшена підтримка та оновлення технології побудови динамічного інтерфейсу.

Недоліком побудови логіки роботи із системою через збережені процедури є те, що складно відстежувати джерело помилок, що можуть виникати при змінах у системі.

Таким чином, веб-сервер та модуль аналізу та оптимізації мають права лише на виклик процедур, необхідних їм у роботі. Кожний з цих модулів має окремого користувача із призначеними їм правами.

3.5.3. Схема побудови динамічного інтерфейсу (база даних)

Із боку бази даних схему побудови динамічного інтерфейсу можна побачити на рис. 15.

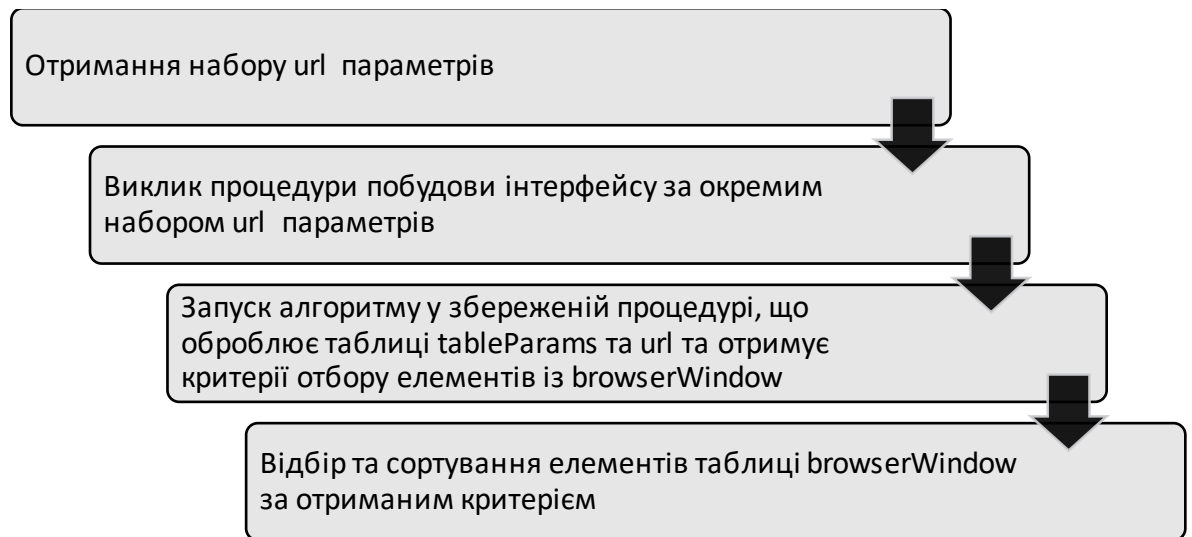


Рис. 15. Схема побудови динамічного інтерфейсу із боку баз даних

3.6. Опис розробленої системи. Інтерфейс

3.6.1. Зовнішній вигляд інтерфейсу

Інтерфейс у розробленій системі являє собою веб-сторінку із елементами керування. Загальний вигляд інтерфейсу можна побачити на рис. 16.

Результати аналізу зображення та знайдених влучень знаходяться у таблиці. Для перегляду доступні номер влучення, номер мішені у якій винайдено влучення, віддаленість від центру та час влучення.

За допомогою списків та кнопок на панелі управління користувач може налагодити отримання саме тій інформації, що його цікавить. Також можна розпочати нову серію у яку будуть записуватись влучення.

На навігаційному мені, що знаходиться зверху веб-сторінки користувач може переглянути довідку про систему.

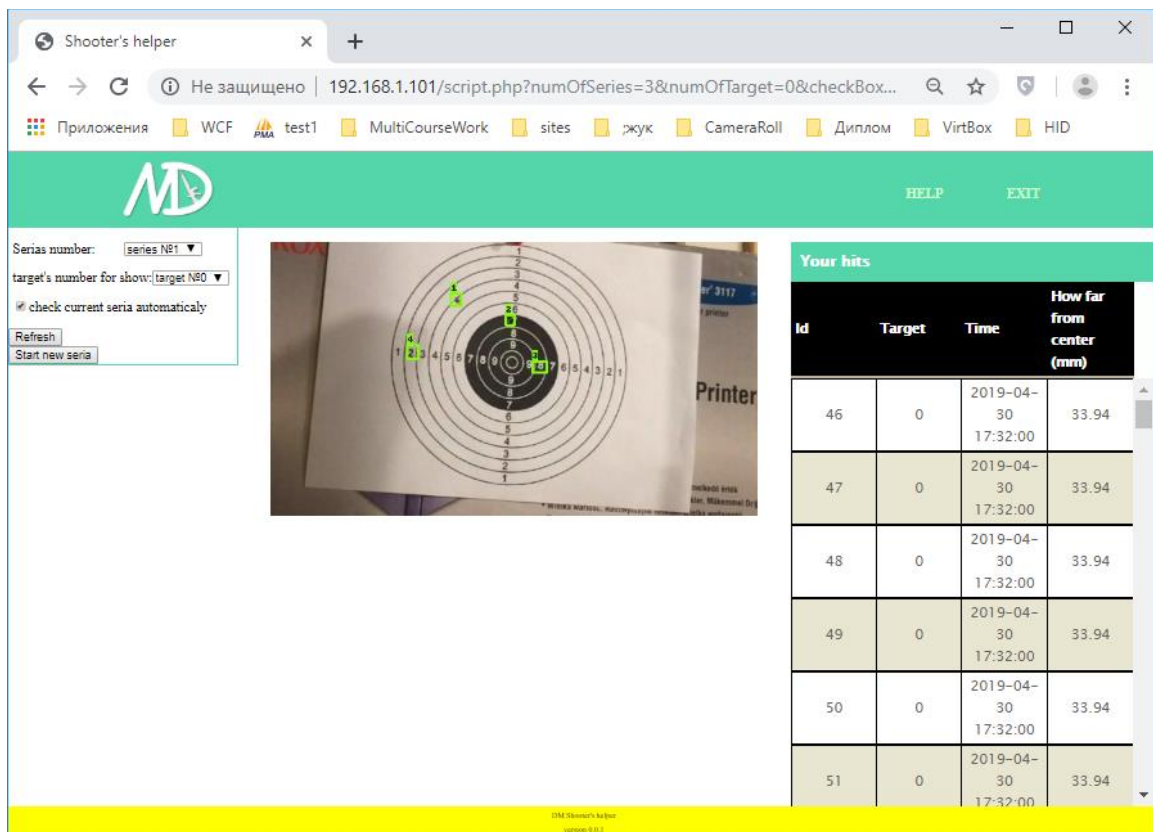


Рис. 16. Інтерфейс системи

3.6.2. Внутрішня структура інтерфейсу

Інтерфейс побудований із використанням мов розмітки CSS, HTML та скриптової мови PHP.

Усі елементи мов розмітки розміщені усередині бази даних у вигляді шаблонів та кешу. На веб-сервері знаходяться PHP скрипти, що викликаються коли користувач вводить правильний IP серверу і знаходиться у мережі системи.

Задача PHP скриптів полягає у тому, щоб встановити з'єднання із базою даних, відіслати параметри та отримати інтерфейс, побудований за цими параметрами. Наведено схему побудови інтерфейсу із боку веб-серверу на рис. 17.

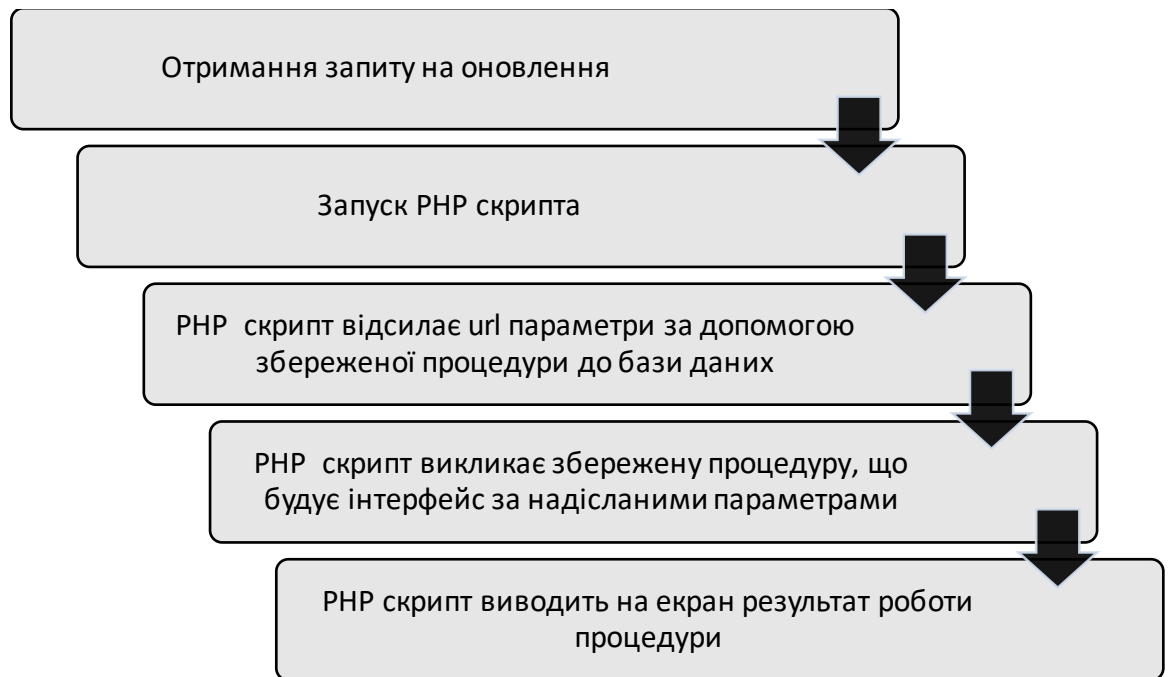


Рис. 17. Схема побудови динамічного інтерфейсу із боку веб-серверу

Для користувача робота із даним інтерфейсом виглядає як робота із звичайною веб-сторінкою. Усі взаємодії користувача із інтерфейсом аналізуються та надсилаються до бази даних за допомогою PHP скрипта. У базі даних відповідно до отриманих даних збирається нова сторінка та надсилається користувачеві.

3.7. Висновки

У даному розділі було розглянуто розроблену систему. Проведений загальний опис системи із прикладами її роботи.

Наведено та проаналізовано зовнішню та внутрішню архітектуру системи.

Вказані варіанти використання системи з боку звичайних користувачів, та варіант використання комунікативного модулю системи з боку модулю аналізу та оптимізації.

Розглянутий спосіб динамічного побудови інтерфейсу та вказані загальні принципи що використовувались при його побудові.

Оглянута розроблена база даних, наведена її схема та проведений огляд її об'єктів та способу взаємодії із базою даних.

Оглянутий розроблений інтерфейс до системи та технології що використовувались під час його розроблення.

4. АНАЛІЗ ТА ТЕСТУВАННЯ РОЗРОБЛЕНОЇ СИСТЕМИ

4.1. Загальний аналіз розробленої системи

Розроблена система вирішує багато проблем тренувань стрільців. Вона дозволяє зменшити витрати часу на організаційні елементи тренувань, такі як:

- заміна мішеней. Система дозволяє запобігти необхідності частій заміні мішеней;
- відмітка влучень. Людині немає необхідності підходити до мішеней та відмічати власні влучення перед початком стрільб наступного стрільця.

Також система покращує безпеку тренувань, так як немає необхідності підходити до мішеней перед стрільбами кожного наступного стрільця. За рахунок цього зменшується частота знаходження людини на вогневому рубежі.

Система дозволяє аналізувати та зберігати влучення, що були знайдені системою. Це надає додаткову інформацію для тренерів та підвищує ефективність тренувань.

Завдяки способам реалізації, доступ до інтерфейсу, а отже – і до системи, можна отримати із будь-якого пристрою, що має браузер для отримання веб-інтерфейсу.

4.2. Тестування розробленої системи

Тестування – це важливий етап створення як програмних, так і програмно-апаратних систем. Воно виявляє проблеми та помилки у реалізації, допомагає продукту бути надійним та задовольняти вимогам.

Тестування у розробленій системі проводилося поетапно. Під час розроблення кожна нова функція, що була добавлена до системи, була протестована окремо.

Після завершення загального циклу розроблення система пройшла загальне тестування, яке перевірило, що можливості системи задовольняє вимогам очікуванням від системи.

Останнім етапом тестування було тестування продуктивності роботи системи.

4.2.1. Процес загального тестування системи

Загальне тестування системи проводилося для виявлення помилок у системі під час її звичайного робочого циклу.

Загальне тестування системи було розділено на 2 частини:

- тестування інтерфейсу із боку користувача системи;
- тестування бази даних із боку модулю аналізу та оптимізації.

Обидві частини процесу загального тестування проводилося із використанням мануального тестування.

Результати тестування інтерфейсу можна побачити на табл. 1.

Таблиця 1

Тестування інтерфейсу

Функція	Дії користувача	Очікуваний результат	Фактичний результат
Надання інтерфейсу у браузері пристрою	Введення IP серверу у адресний рядок	Отримання початкової сторінки інтерфейсу	Було отримано початкову сторінку інтерфейсу
Зміна серії пострілів, інформація щодо якої отримується	Зміна обраного елемента у списку серій пострілів	Зміна наповнення таблиці із пострілами	Була змінено наповнення таблиці, що отримується у інтерфейсі

Продовження табл. 1

Зміна фотографії мішені, що отримується у інтерфейсі	Зміна обраного елемента у списку мішеней	Зміна фотографії, що отримується браузером	Була змінена фотографія мішені на обрану із списку
Оновлення сторінки	Натискання на кнопку оновлення сторінки або оновлення сторінки за допомогою браузера	Сторінка оновлюється, на сторінці наявна остання за актуальністю інформація	Сторінка була оновлена, була отримана інформація що враховує останні оновлення у базі даних
Додання серії пострілів	Натискання на кнопку, що додає серії пострілів	Додається нова серія до списку серій. Після оновлення сторінки список серій у інтерфейсі оновлюється	При натисканні на кнопку до системи була додана нова серія. Після оновлення сторінки список серій, що наявний у інтерфейсі був оновлений
Влучення додаються до системи	Створення влучення у мішені	Додається влучення до серії, що враховує час створення влучення	При додаванні влучень, влучення додавались до правильної за часом серії пострілів

Тестування бази даних із боку модулю аналізу та оптимізації можна побачити у табл. 2.

Тестування бази даних

Функція	Дії користувача	Очікуваний результат	Фактичний результат
Додання влучень до системи	Виклик збереженої процедури на додання влучення	До системи додається влучення у серію, список влучень у інтерфейсі змінюється	До системи було додано влучення у серію з урахуванням часу пострілу, у інтерфейсі можна переглянути дане влучення
Додання мішені до системи	Виклик збереженої процедури, що додає мішень	До системи додається мішень, список мішеней у інтерфейсі змінюється	У системі додалася мішень. Список мішеней у інтерфейсі змінився
Додання фото мішені до системи	Виклик збереженої процедури, що додає фото	До системи додається місцезнаходження фото мішені. Веб-інтерфейс показує фото мішені	До системи додалося фото мішені. Веб-інтерфейс показує фото мішені за необхідністю
Отримання інформації о влученнях у системі за номером серії	Виклик збереженої процедури, що повертає влучення, збережені у системі за параметром номеру серії	Процедура повертає інформацію о влученнях збережених у системі за номером серії	Процедура повернула інформацію о влученнях за номером серії, що була вказана під час виклику збереженої процедури

За допомогою даних тестувань була перевірена правильність роботи системи, зокрема роботу інтерфейсу під час роботи із ним користувача та роботу бази даних під час її взаємодії із модулем, що виконує пошук змін у стані мішеней.

4.2.2. Процес тестування продуктивності роботи системи

Тестування продуктивності роботи системи необхідно для того, щоб упевнитись у можливостях системи працювати із групами користувачів та з декількома камерами одночасно.

Для перевірки роботи бази даних було проведено тестування, що включало у себе додання великої кількості влучень до системи у малий проміжок часу. У результаті помічено суттєве уповільнення роботи системи. Було вирішено переробити та оптимізувати алгоритм, що займався обробкою та аналізом влучень, що додаються до системи. При повторному тестуванні помічено покращення у швидкодії системи.

Для перевірки роботи інтерфейсу було проведено тестування із одночасним підключенням до веб-серверу деякої кількості користувачів, зокрема – 10, 50, 100. Помічене несуттєве сповільнення роботи на кількості користувачів 50 та менше, та суттєве при підключенні більшої кількості користувачів. Також робота бази даних сповільнюється, якщо до системи одночасно звертаються багато користувачів. Дана проблема може бути вирішена при використанні на ролі серверів більш потужних пристроїв.

4.3. Порівняння із існуючими продуктами

Порівняння із існуючими продуктами дозволяє виявити недоліки, що існують у системі, для їх можливого виправлення у подальших версіях, та відмітити переваги над існуючими системами.

Розроблена система усуває такі недоліки існуючих рішень:

- Одночасне використання. Розроблена система дозволяє використовувати її майже будь – якої кількості користувачів водночас.
- Не потребує допоміжних пристроїв. У розробленій системі немає необхідності використання допоміжних пристроїв на зброї.

- Підтримка великої кількості платформ. Завдяки використанню веб-інтерфейсу, майже будь-який пристрій користувачів, підключений до локальної мережі, може отримати доступ до інтерфейсу.
- Простота розгортки. Система, на відміну від більшості інших, не потребує тонких налаштувань при її розгортці.
- Гнучкість використання. Система може працювати із будь-якою зброєю, що робить отвори у полотні мішені.
- Легкість змін у системі. Завдяки використаним технологічним рішенням, інтерфейс та сутність системи може легко змінюватись під нові вимоги користувачів.

Розроблена система також має певні недоліки, що можуть бути усунені у наступних версіях:

- Відсутність збору інформації о стані і положенні зброї.
- Відсутність реєстрації у системі. В користувачів немає власного кабінету та статистики відсортованої та проаналізованою за користувачем.
- Система не може працювати із «сухими» пострілами. Система вимагає утворення отвору у мішені при влученні.

Отже, система має певні переваги та недоліки над існуючими рішеннями. Переваги є суттєві покращеннями у можливостях та гнучкості системи. Недоліки можна усунути під час подальшої роботи та вдосконалення системи.

4.4. Шляхи вдосконалення та змін системи

Розроблена система може бути вдосконалена, отже необхідно проаналізувати можливі кроки для покращення можливостей та усунення недоліків у системі.

4.4.1. Покращення та збільшення функціональних можливостей у розробленій системі

Вдосконалення розробленої системи може проводитися у кількох напрямках: виправлення недоліків системи та додання можливостей до системи.

Для виправлення недоліків у системі необхідно додати до системи модулі та частини що їх виправляють. У попередньому підрозділі були зазначені недоліки системи відносно існуючих рішень, а отже, для покращення системи можна розпочати із їх виправлення:

- Додати авторизацію користувача. Додання окремого доступу для кожного користувача дозволить аналізувати результати роботи системи по окремому користувачу. Збирання статистики та аналіз стрільб для окремої людини значно підвищить ефективність використання системи та дозволить слідкувати за покращеннями результатів окремого стрільця, та будувати програму тренувань для кожного окремо з урахуванням даних, що зберігаються у системі.
- Додання статистичного аналізу до системи. Дана функція, разом із функцією авторизації користувача, допоможе як тренерам слідкувати за результатами учнів, так і учням слідкувати за своїм прогресом та недоліками.

Також вдосконалити розроблену систему можна шляхом збільшення можливостей інтерфейсу, а саме:

- Додання функції побудови вправ. Додати у систему можливості будування вправи для виконання стрільцем. Це дасть змогу тренеру планувати тренування заздалегідь, прискорить процес пояснення задачі, що ставиться перед стрільцем.
- Додання режиму планування занять. Тобто щоб користувачі, що відповідають за тренування, а саме – тренери, мали змогу планувати послідовність вправ що необхідно пройти стрільцю. Відповідно система повинна стежити за вдалістю виконання цього набору та

збирати статистику, аналізувати її. Така функція має враховувати набір мішеней, що зараз доступний у системі.

- Додання режиму проведення змагань. На спортивних змаганнях із стрільб організатори та тренери стикаються з тими ж самими проблемами що і на тренуваннях. За допомогою такого режиму можна будувати план змагань із поясненнями, що дозволить зменшити навантаження на тренерів та дозволить стрільцям готуватися до вправ.
- Додання функцій щодо слідкування за станом системи, у якій тренуються стрільці. Сигналізувати про необхідність заміни мішеней, відслідковувати їх стан.

Отже, розроблена система має багато варіантів для вдосконалення своєї роботи та збільшення можливостей, що вона надає.

4.4.2. Зміна напрямлення роботи системи

Завдяки технологіям, що були використані під час розроблення системи, можливо у короткий термін змінити напрямлення роботи системи.

Використання системи визначення змін у стані об'єкту можливе у багатьох галузях, як наукових, так і промислових.

Використання системи у промисловості може допомогти відслідковувати зміни у стані механізмів, пристроїв, що повинні зоставатися незмінними, або навпаки – відслідковувати зміни, і перевіряти їх схожість до очікуваних.

Використання системи у науковій сфері може бути дуже різноманітним. За допомогою системи визначення змін у стані об'єкту можна досліджувати мікробіологію, хімію. Також можна досліджувати космос, знаходити нові астероїди та інші незареєстровані об'єкти.

Технології, що використовувались під час розроблення баз даних та інтерфейсу дозволяють систему зробити гнучкою до змін. Інтерфейс системи можна кардинально змінити за допомогою змін у шаблонах, що

знаходяться у базі даних. База даних побудована так, що для використання системи у інших галузях необхідно додати таблиці для збереження інформації, що отримується із модулю відстеження змін у стані об'єктів, та змінити алгоритми обробки інформації.

4.5. Висновки

У даному розділі система була проаналізована із різних боків.

Був проаналізований загальний вигляд та можливості системи, що отримує користувач під час її використання.

Були відмічені переваги та недоліки даної системи відносно існуючих на ринку систем, що вирішують проблеми стрільців та їх тренувань.

Був описаний процес тестування системи та результати, що були отримані під час тестування.

Були проаналізовані можливі шляхи для вдосконалення системи, були відмічені декілька варіантів для вдосконалення та функцій, що спростять роботу із системою та збільшать ефективність.

ВИСНОВКИ

Метою даного дипломного проекту було створення комунікативного модулю системи динамічного визначення змін у стані об'єкту. Цільова система створена для використання стрільцями, а саме – для визначення змін у стані мішеней.

Проведено аналіз існуючих систем, що спрощують процес тренувань стрільців, зроблено висновки щодо можливостей, що надають такі системи, та недоліків. Розроблені вимоги до системи, що враховують як цілі, що переслідують стрільці при тренуваннях, так і недоліки та переваги існуючих систем.

Проаналізовані можливі засоби реалізації та обрані ті, що надають можливість у простий спосіб дотриматись вимог та створити комунікативний модуль до системи.

Розроблений комунікативний модуль надає такі можливості:

- взаємодія із іншими модулями, із яких складається система;
- надання інтерфейсу користувачеві, що надає можливості керування системою та перегляду результатів її роботи;
- зберігання даних о роботі як системи загалом, так і окремих її частин;
- обробка даних, що отримані із інших модулів, для надання користувачеві актуальної та простої для аналізу інформації;
- проста зміна інтерфейсу та алгоритмів для обробки інформації для задоволення вимог різних груп користувачів.

Комунікативний модуль виконує задачі, що поставлені при його проектуванні та задовольняє вимогам. Він дозволяє взаємодіяти із системою користувачам та іншим модулям, із яких складається система.

Розроблена система дозволяє стрільцям підвищити ефективність своїх тренувань, підвищити безпеку та зручність роботи із мішенями.

Система розроблена із гнучкою структурою. Це дозволяє використовувати загальну схему побудови як системи загалом, так і окремо комунікативного модулю. Отже дану систему та її окремі частини можна швидко трансформувати для застосування у інших галузях та напрямках. Також система має перспективи щодо розвитку у рамках обраного напрямку використання шляхом додання функціонала до окремих її частин.

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Коваленко О. Є. СТАНДАРТИЗАЦІЯ ФОРМАЛЬНОГО ОПИСУ СИСТЕМНОЇ АРХІТЕКТУРИ [Електронний ресурс] Інститут проблем математичних машин і систем НАН України. – 2015. – Режим доступу до ресурсу: http://conf.atsukr.org.ua/conf_files/conf_dir_24/Kovalenko_sppr2015.pdf.
2. Cockburn A. Writing Effective Use Cases / Allistair Cockburn., 2000.
3. Ryan S. Sams Teach Yourself SQL in 24 Hours / S. Ryan, P. Ron, D. Arie., 2008. – (Sams Teach Yourself -- Hours).
4. MantisX Shooting Performance System [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://mantisx.com/products/mantisx>.
5. SCATT. Стрілецькі тренажери [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://www.scatt.ru/scatt-mx-02>.
6. Marksman training system [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://www.marksman.se/>.
7. David M. K. Теория и практика построения баз данных / М. Kroenke David., 2005.
8. Що таке NoSQL [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://aws.amazon.com/ru/nosql/>.
9. Орлов С. А. Программная инженерия. Технологии разработки программного обеспечения / Сергей Александрович Орлов., 2016.
10. Порівняння NoSQL систем управління базами даних [Електронний ресурс]. – 2014. – Режим доступу до ресурсу: <http://devacademy.ru/posts/nosql/>.
11. Database from Oracle [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: <https://www.oracle.com/database/>.
12. Installing Oracle Database and Creating a Database [Електронний ресурс]. – 2019. – Режим доступу до ресурсу:

https://docs.oracle.com/cd/E11882_01/server.112/e10897/install.htm#ADMQS022.

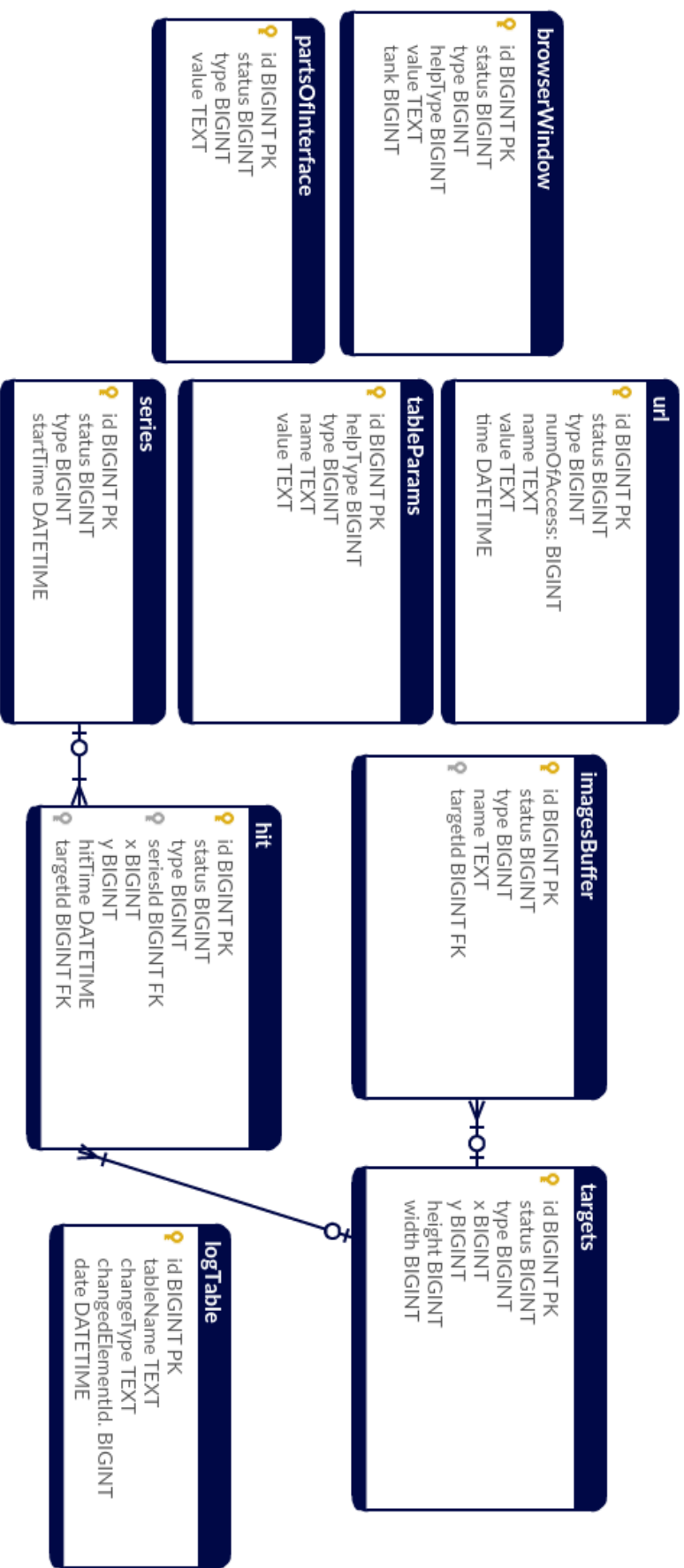
13. Introduction to the Oracle Database [Электронный ресурс]. – 2019. – Режим доступа до ресурсу: https://docs.oracle.com/cd/B13789_01/server.101/b10743/intro.htm.
14. Advantages & Disadvantages of Oracle SQL [Электронный ресурс]. – 2015. – Режим доступа до ресурсу: <https://www.techwalla.com/articles/advantages-disadvantages-of-oracle-sql>.
15. Кайт Т. Oracle для профессионалов. Архитектура и основные особенности / Том Кайт., 2013.
16. Петкович Д. Microsoft SQL Server. Руководство для начинающих / Душан Петкович., 2013.
17. Advantages & Disadvantages of Microsoft SQL [Электронный ресурс]. – 2012. – Режим доступа до ресурсу: <https://www.techwalla.com/articles/advantages-disadvantages-of-microsoft-sql>.
18. What is PostgreSQL? [Электронный ресурс]. – 2019. – Режим доступа до ресурсу: <https://www.postgresql.org/about/>.
19. Введения в MySQL. Установка сервера. [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: <https://metanit.com/sql/mysql/1.1.php>.
20. What is MySQL [Электронный ресурс]. – 2019. – Режим доступа до ресурсу: <https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html>.
21. Why MySQL [Электронный ресурс]. – 2019. – Режим доступа до ресурсу: <https://www.mysql.com/why-mysql/>.
22. What is SQLite [Электронный ресурс]. – 2019. – Режим доступа до ресурсу: <https://www.sqlite.org/about.html>.
23. Що таке інтерфейс? Різновиди інтерфейсів [Электронный ресурс] – Режим доступа до ресурсу: <https://anisim.org/articles/chto-takoe-interfeys/>.

24. 7 unbreakable laws of user interface design [Електронний ресурс]. – 2014.
– Режим доступу до ресурсу: <https://99designs.com/blog/tips/7-unbreakable-laws-of-user-interface-design/>.
25. Види інтерфейсів користувача операційних систем [Електронний ресурс] – Режим доступу до ресурсу: <https://infl.info/user-interface>.

ДОДАТКИ

ДОДАТКИ

Додаток 1
Копії графічних матеріалів



ДП.045200-06-99

Автоматизована система динамічного
визначення змін у стані об'єкта.
Комунікативний модуль.
Схема бази даних комунікативного модулю.
ER діаграма



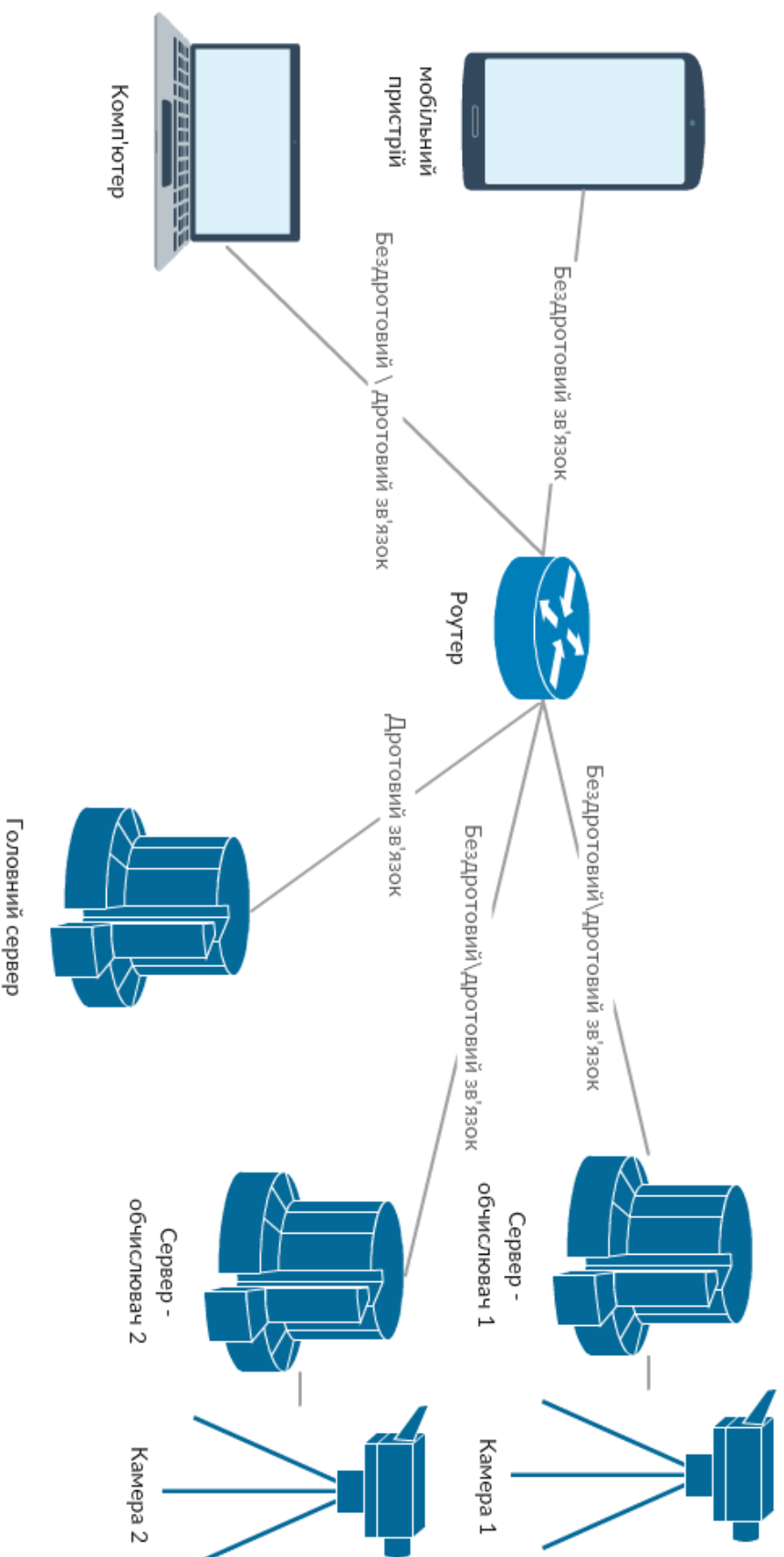
ДП.045200-07-99

Автоматизована система динамічного визначення змін у стані об'єкта. Комунікативний модуль.

Варіанти використання комунікативного модулю.

Діаграма варіантів використання

Зовнішня структура системи



Охримчук Д.Д., група КІП-51

Інтерфейс системи

Shooter's helper


Не захищено | 192.168.1.101/script.php?numOfSeries=3&numOfTarget=0&checkBox=...

ПриложенияWCFtest1MultiCourseWorksitesжукCameraRollДипломVirtBoxHID

MD

HELPEXIT

Series number:Series №1▼
target's number for show:target №0▼
☒ check current seria automatically
Refresh
Start new seria



Your hits			
Id	Target	Time	How far from center (mm)
46	0	2019-04-30 17:32:00	33.94
47	0	2019-04-30 17:32:00	33.94
48	0	2019-04-30 17:32:00	33.94
49	0	2019-04-30 17:32:00	33.94
50	0	2019-04-30 17:32:00	33.94
51	0	2019-04-30 17:32:00	33.94

Охримчук Д.Д., група КП-51

Додаток 2

Текст програми

2.1. PHP скрипт для отримання стартової сторінки інтерфейсу:

```
<?php
    $mysqli = new
mysqli("localhost","u1","13542Den","TestDatabase_1");
    $res = $mysqli->query("CALL `GetSite`(0);");
    while($row = mysqli_fetch_assoc($res))
    {
        echo $row['value'];
    }
    $mysqli->close();
?>
```

2.2. PHP функція для додання URL до бази:

```
function addToUrl($mysqliFun,$urlName, $urlHelpNum)
{
    if(!$mysqliFun->query("CALL
`addUrl`(0,``.$urlName.``,`.$_GET[$urlName].`,`.$urlHelpNum.``);")){
        echo "Не удалось вызвать хранимую процедуру: (" . $mysqliFun-
>errno . ") " . $mysqliFun->error;
        echo "vse ploho l";
        echo $helpNum;
    }
}
```

2.3. PHP скрипт, що завантажує URL до бази даних та отримує відповідний інтерфейс:

```
$row = mysqli_fetch_assoc($mysqli->query("SELECT `getUrlNumAccess`()
AS `getUrlNumAccess`");
$helpNum=$row['getUrlNumAccess'];
addToUrl($mysqli,"", $helpNum);
addToUrl($mysqli,"numOfSeries", $helpNum);
addToUrl($mysqli,"numOfTarget", $helpNum);
addToUrl($mysqli,"other", $helpNum);
```

```
$res = $mysqli->query("CALL `GetSite`('.$helpNum.')");  
while($row = mysqli_fetch_assoc($res))  
{  
    echo $row['value'];  
}
```

2.4. Шаблон PHP скрипта, що завантажує фотографію із бази даних (BLOB type):

```
<?php  
$db = mysqli_connect("localhost","u1","13542Den","TestDatabase_1"); //keep  
your db name  
$sql = "SELECT value FROM image WHERE id = 1";  
$sth = $db->query($sql);  
$result=mysqli_fetch_array($sth);  
echo '<img src="data:image/jpeg;base64,'.base64_encode( $result['value']  
?>
```

2.5. Код SQL процедури, що створює елемент інтерфейсу для пострілу за його ідентифікатором

updatebrowserWindowByNewHit(newId)

```
INSERT INTO
browserwindow
VALUES (getNewIdForBrowserWindow(),
      0,
      getHitType(),
      getHelpTypeForHitBySeries(getSeriesByTime((SELECT hitTime FROM
hit WHERE id=newId))),
      (SELECT REGEXP_REPLACE(
        (SELECT value FROM partsoftheinterface WHERE type=0),
        '@1',
        CONCAT(CONCAT(CONCAT(
          (SELECT REGEXP_REPLACE(
            (SELECT value FROM partsoftheinterface WHERE
            type=1),
            '@1',
            getHitNumInSeria(newId))),
            (SELECT REGEXP_REPLACE(
              (SELECT value FROM
              partsoftheinterface WHERE
              type=1),
              '@1',
              (SELECT targetId
              FROM hit WHERE id=newId)))),
            (SELECT REGEXP_REPLACE(
              (SELECT value FROM partsoftheinterface
              WHERE type=1),
              '@1',
              (SELECT hitTime FROM hit WHERE
              id=newId)))),
            (SELECT REGEXP_REPLACE(
              (SELECT value FROM partsoftheinterface
              WHERE
              type=1),
              '@1',
              getToCenterLengthByHitId(newId)))))),
      3000)
```

2.6. Код SQL функції, що створює елемент інтерфейсу для пострілу за його ідентифікатором:

getHitNumInSeria (BIGINT newHitId)

```
BEGIN
SET @rownumber = 0;
RETURN (SELECT num FROM (SELECT (@rownumber:=@rownumber+1)
as num,id FROM hit WHERE seriesId=(SELECT seriesId FROM hit WHERE
id = newHitId) ORDER BY id) as a WHERE id = newHitId);
END
```

2.7. Код SQL процедури, що повертає інтерфейс користувачеві за відповідними URL параметрами:

getSite (AccessId)

```
BEGIN
CREATE TEMPORARY TABLE TempTable(helpType BIGINT);
INSERT INTO TempTable(helpType) (SELECT helpType FROM
TableParams
INNER JOIN url ON url.name=TableParams.name AND
url.value=TableParams.value AND url.numOfAccess=newNumOfAccess);
INSERT INTO TempTable(helpType) (SELECT helpType FROM
TableParams
WHERE TableParams.name NOT IN (SELECT name FROM url));
SELECT * FROM browserwindow WHERE helpType IN (SELECT helpType
FROM TempTable)
ORDER BY rank;
END
```

2.8. Код SQL функції, що розраховує віддаленість влучення від центру мішені:

getToCenterLengthByXYHWnewXnewY(x, y, h, w, newX, newY, roundCount)

```
RETURN ROUND((SQRT((x+(w/2)-newX)*(x+(w/2)-newX)+(y+(h/2)-newY)*(y+(h/2)-newY))), roundCount)
```

2.9. Код SQL триггеру, що заповнює таблицю змін у базі даних при доданні нового влучення.

TrigerInsertHitTable

```
INSERT INTO LogTable (tableName, ChangeType, ChangedElementId, date)  
VALUES ('hit', 'insert', NEW.id, Now())
```

2.10. Код SQL триггеру, що створює елемент інтерфейсу при доданні нової серії

updateDataBaseByNewSeries

```
BEGIN  
INSERT INTO browserwindow (id,status,type,helpType,value,rank)  
VALUES (getNewIdForBrowserWindow(), 1, 1, -1,(SELECT  
REGEXP_REPLACE((SELECT REGEXP_REPLACE((SELECT value  
FROM partsoftheinterface WHERE  
type=2),'@1',NEW.id)),'@2',CONCAT("series №",NEW.id))),  
getNewRankForSeries());  
INSERT INTO TableParams(id,type,helpType,name,value) VALUES  
(getNewIdForTableParams(),getSeriesType(), getNewHelpType(),  
"numOfSeries", NEW.id);  
END
```

2.11. Код SQL триггеру, що створює елемент інтерфейсу при доданні нової фото

updateBrowserWindowByImagesBuffer

```
INSERT INTO  
browserwindow  
VALUES (getNewIdForBrowserWindow(), 0,  
getPhotoType(),getHelpTypeForPhotoByTargetId((SELECT targetId FROM  
imagesBuffer WHERE id = newId)),(SELECT  
REGEXP_REPLACE((SELECT value FROM partsoftheinterface WHERE  
type=3),'@1', (SELECT name FROM imagesBuffer WHERE id =  
newId))),6500)
```

2.12. Код SQL функції, що повертає за номером влучення віддаленість до центру

getToCenterLengthByHitId (hitId)

```
RETURN getToCenterLengthByXYHWnewXnewY(  
(SELECT x FROM targets WHERE targets.id=(SELECT targetId FROM hit  
WHERE id=hitId)),  
(SELECT y FROM targets WHERE targets.id=(SELECT targetId FROM hit  
WHERE id=hitId)),  
(SELECT height FROM targets WHERE targets.id=(SELECT targetId FROM  
hit WHERE id=hitId)),  
(SELECT width FROM targets WHERE targets.id=(SELECT targetId FROM  
hit WHERE id=hitId)),  
(SELECT x FROM hit WHERE id = hitId),  
(SELECT y FROM hit WHERE id = hitId),  
2)
```

2.13. Код SQL триггеру, що створює елемент інтерфейсу при доданні нової серії

updateDataBaseByNewSeries


```

BEGIN
INSERT INTO browserwindow(id,status,type,helpType,value,rank)
VALUES (getNewIdForBrowserWindow(), 1, 1, -1,(SELECT
REGEXP_REPLACE((SELECT REGEXP_REPLACE((SELECT value
FROM partsoftheinterface WHERE
type=2),'@1',NEW.id)),'@2',CONCAT("series №",NEW.id))),
getNewRankForSeries());
INSERT INTO TableParams(id,type,helpType,name,value) VALUES
(getNewIdForTableParams(),getSeriesType(), getNewHelpType(),
"numOfSeries", NEW.id);
END

```

2.14. Код SQL процедури, що додає до бази даних фото та відповідний до нього номер мішені

addPhoto(newTargetId, newName)

```

BEGIN
set @idForInsert=getNewIdForImagesBuffer();
INSERT INTO
imagesBuffer (id, status, type, name, targetId)
VALUES (@idForInsert,0, getNewTypeForImagesBuffer(),newName,
newTargetId);
CALL updateBrowserWindowByImagesBuffer(@idForInsert);
END

```

2.15. Код SQL функції, що за часом повертає номер серії, що була активна у даний період

getSeriesByTime (time)

```

RETURN (SELECT id FROM series
WHERE startTime<time
ORDER BY startTime DESC
LIMIT 1)

```

2.16. Код SQL функції, що знаходить новий ранг для елементу серії, за яким елемент буде сортуватися відносно інших

getNewRankForSeries

```
RETURN (SELECT MAX(rank)+1  
        FROM browserwindow  
        WHERE rank <= 199)
```

2.17. Код SQL функції, що знаходить порядковий номер влучення у серії пострілів

getHitNumInSeria (newHitId)

```
BEGIN  
SET @rownumber = 0;  
RETURN (SELECT num FROM (SELECT (@rownumber:=@rownumber+1)  
as num,id FROM hit WHERE seriesId=(SELECT seriesId FROM hit WHERE  
id = newHitId) ORDER BY id) as a WHERE id = newHitId);  
END
```

2.18. Код SQL функції, що знаходить новий ID для елементу у таблиці для збереження кешу інтерфейсу

getNewIdForBrowserWindow

```
RETURN (SELECT MAX(id)+1 FROM browserwindow)
```

2.19. Код SQL функції, що повертає сформований для влучення елемент інтерфейсу

functionHitTableText (hitId)

```
RETURN (SELECT REGEXP_REPLACE((SELECT value FROM  
partsoftheinterface WHERE type=0),'@1', hitId))
```

2.20. Код SQL функції, що повертає ID типу мішені. За таким принципом можна узнати типи різних елементів, що зберігаються у базі

`getTargetType`

`RETURN 27`

2.20. Код SQL процедури, що повертає інформацію о мішенях

getCameraTargetOptions

```
SELECT identificator, x, y, height, width, targets.id  
FROM camera  
INNER JOIN targets ON camera.id=targets.cameraId  
WHERE camera.sessionId=getCurrentSessionId()
```

Додаток 3
Копія презентації

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО”



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

**АВТОМАТИЗОВАНА СИСТЕМА ДИНАМІЧНОГО
ВИЗНАЧЕННЯ ЗМІН У СТАНІ ОБ'ЄКТА.
КОМУНІКАТИВНИЙ МОДУЛЬ**

Виконав: Охримчук Денис Дмитрович

Науковий керівник: ст. викладач, к.т.н., Люшенко Леся Анатоліївна

Київ – 2019

ПОСТАНОВКА ЗАДАЧІ

Мета проекту: розробити програмну систему для динамічного визначення змін у стані об'єкту. У ролі об'єкту розглядається мішень для стрільб.

Завдання:

1. Проаналізувати існуючі рішення.
2. Розробити комунікативний модуль до системи.
3. Провести тестування розробленого модулю та системи у цілому.
4. Проаналізувати роботу та перспективи розвитку системи.

АКТУАЛЬНІСТЬ

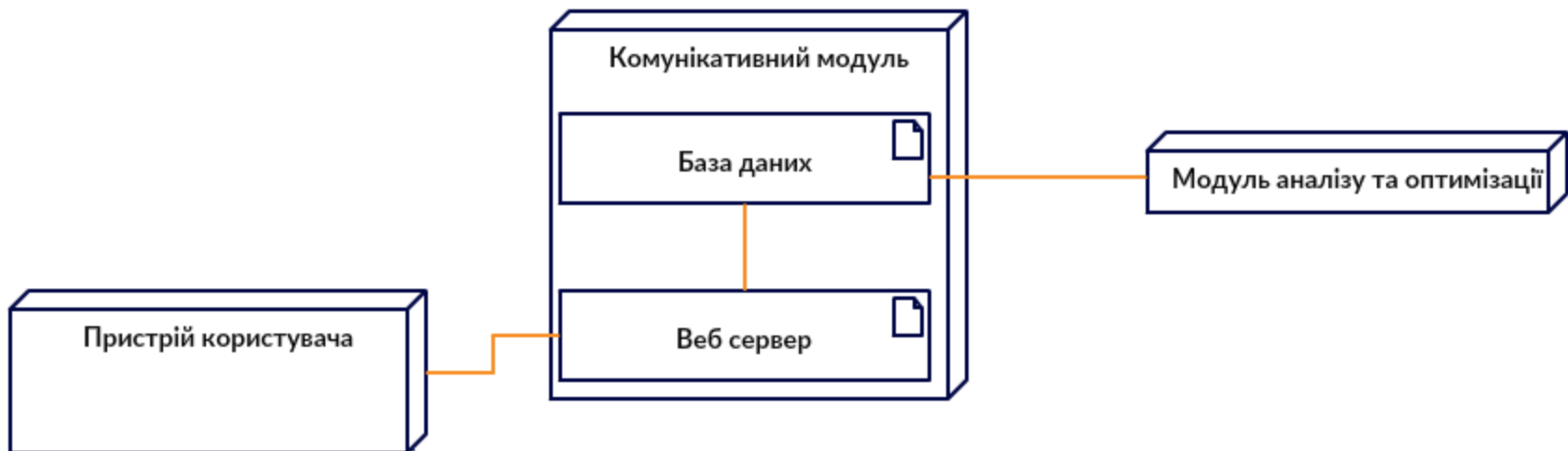
Сфера застосування розробки: робота та аналіз стану мішеней. Зараз витрачається багато зусиль та часу на отримання даних із мішеней під час стрільб.

Альтернативні сфери застосування системи динамічного визначення змін у стані об'єкту:

- Медицина
- Астрономія
- Промисловість
- Охоронні системи

АРХІТЕКТУРА СИСТЕМИ

Внутрішня архітектура має вигляд:



ЗАСОБИ РОЗРОБКИ

Розробка бази даних.

База даних була розроблена засобами СКБД MySQL.

СКБД MySQL був обраний за критеріями:

- Безкоштовна для використання.
- Проста у розгортці та підтримці.
- Проста у використанні.
- Швидкодія та ефективність на високому рівні.

ЗАСОБИ РОЗРОБКИ

Розробка інтерфейсу.

Реалізація інтерфейсу була обрана на основі веб-інтерфейсу із певними модифікаціями.

Перевагами даного рішення є:

- Просте забезпечення кросплатформності.
- Відсутня необхідність встановлення програмного забезпечення на пристрої користувачів.
- Просте оновлення та модифікація інтерфейсу

РОЗРОБЛЕНА СИСТЕМА. КОМУНІКАЦІЙНИЙ МОДУЛЬ



Комунікаційний модуль у системі має наступне призначення:

- Взаємодія між модулями.
- Надання користувацького інтерфейсу.
- Збереження даних.
- Обробка збережених даних.

РОЗРОБЛЕНА СИСТЕМА. ІНТЕРФЕЙС

Інтерфейс системи є підмодулем комунікаційного модулю.

Інтерфейс надає користувачу доступ до системи та можливості для її контролю.

Особливостями модулю інтерфейсу системи є те, що він будується динамічно за запитами до бази даних, що дозволяє гнучко змінювати інтерфейс відповідно до вимог користувача.

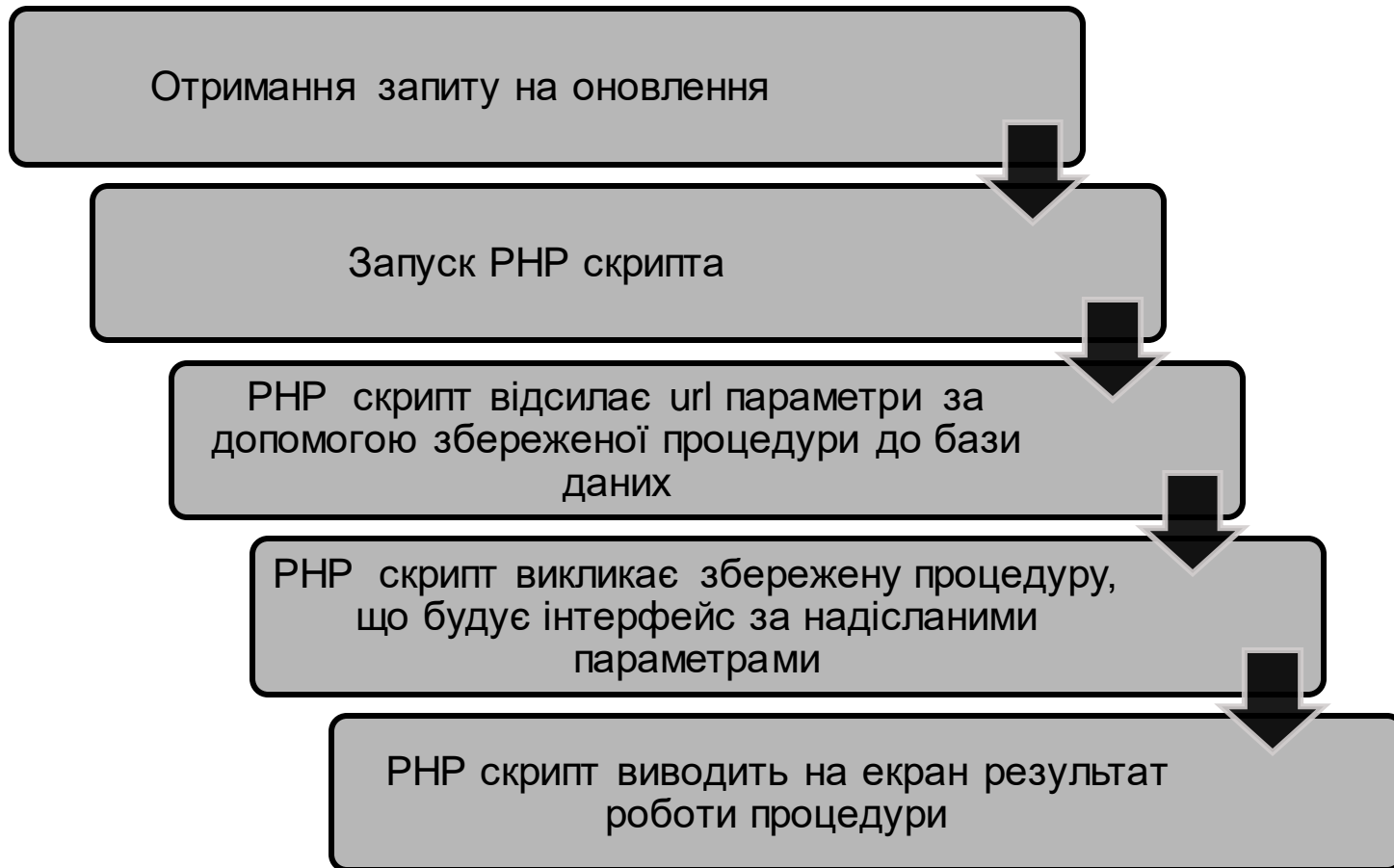
РОЗРОБЛЕНА СИСТЕМА. БАЗА ДАНИХ

База даних системи є підмодулем комунікаційного модулю.

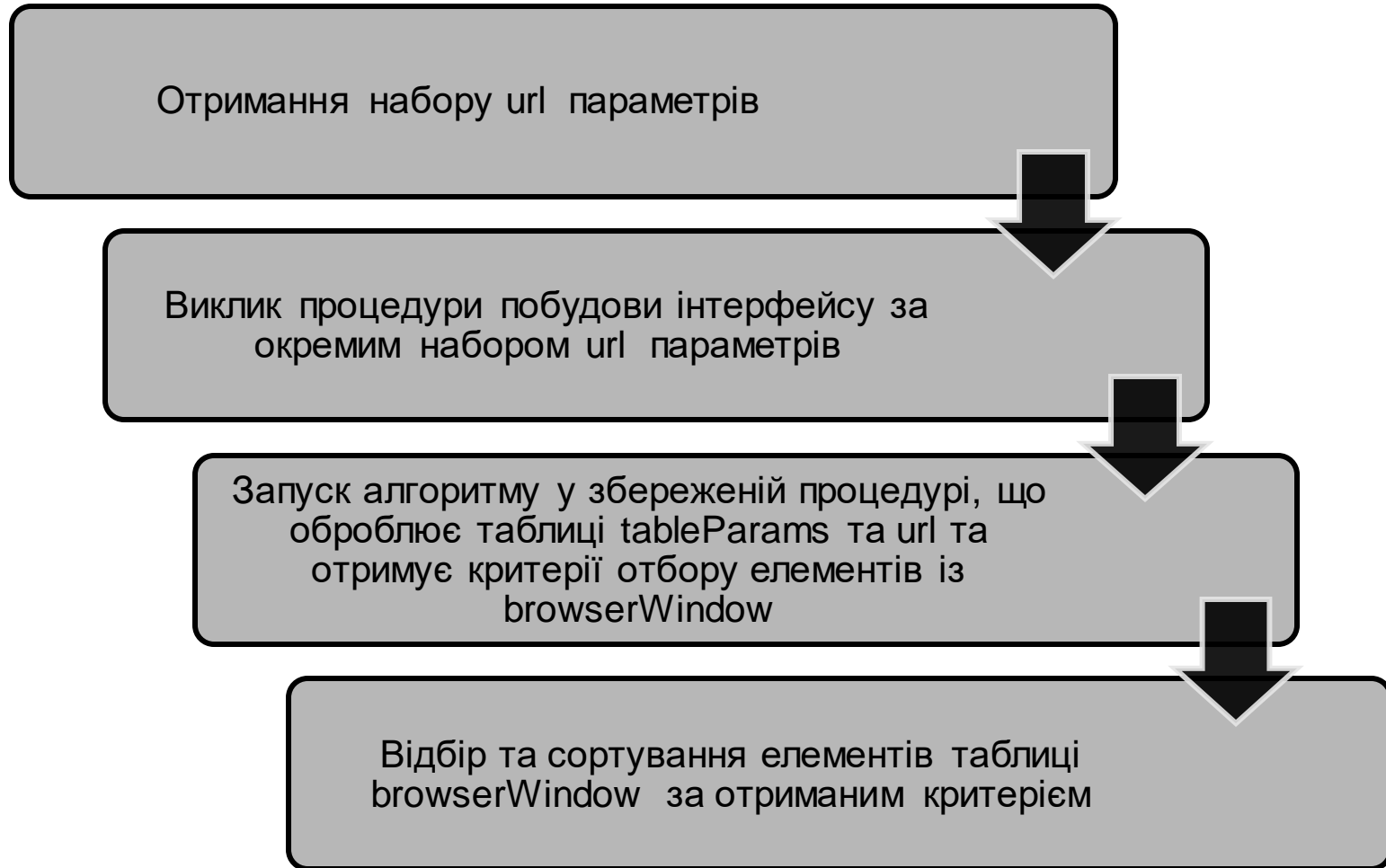
База даних має наступне призначення у рамках модулю:

- Збереження даних о роботі системи.
- Обробка даних, що отримуються із модулів.
- Підтримка роботи динамічного інтерфейсу.
- Взаємодія між модулями.

РОЗРОБЛЕНА СИСТЕМА. ДИНАМІЧНИЙ ІНТЕРФЕЙС. СХЕМА РОБОТИ ІЗ БОКУ ВЕБ-СЕРВЕРУ



РОЗРОБЛЕНА СИСТЕМА. ДИНАМІЧНИЙ ІНТЕРФЕЙС. СХЕМА РОБОТИ ІЗ БОКУ БАЗИ ДАНИХ



АПАРАТНА ЧАСТИНА

Сервер: Orange Pi 256M

У рамках системи цей сервер необхідний для:

- Розгортки веб-серверу, до якого користувачі мають доступ.
- Розгортка бази даних, до якої мають обмежений доступ окремі модулі системи.

Модуль аналізу та оптимізації взаємодіє із сервером шляхом підключення за допомогою мережевого протоколу SSH.

ТЕСТУВАННЯ МОДУЛЮ

Модуль був протестований на відповідність функціональним вимогам та стабільність роботи під навантаженням.

Модуль відповідає функціональним вимогам, а саме:

- Надає можливості взаємодії поміж модулями.
- Надає інтерфейс користувачеві.
- Зберігає дані, отримані в ході роботи системи.

Модуль показує достатню стабільність роботи при:

- Підключенню до системи <10 користувачів.
- Отримання не більше за 10 влучень у секунду.

КРИТЕРІЇ ОЦІНЮВАННЯ ЯКОСТІ РОЗРОБЛЕНОГО МОДУЛЮ

Розроблений модуль оцінюється за наступними критеріями:

- Функціональність.
- Відмовостійкість.
- Мобільність.
- Швидкодія.
- Модифікованість.
- Кросплатформність.
- Універсальність застосування.

КРИТЕРІЇ ОЦІНЮВАННЯ ЯКОСТІ РОЗРОБЛЕНОГО МОДУЛЮ

Функціональність модулю відповідає вимогам, що були висунуті до нього та його частин. Модуль надає можливості зберігання даних, їх обробки, та надання користувачеві інтерфейсу для взаємодії із системою.

Відмовостійкість забезпечується тим, що усі взаємодії із системою проходять через базу даних, відповідно усі дані зберігаються у надійному середовищі, що дозволяє відновити роботу системи після аварійних ситуацій та при помилках.

КРИТЕРІЇ ОЦІНЮВАННЯ ЯКОСТІ РОЗРОБЛЕНОГО МОДУЛЮ

Мобільність модулю на високому рівні, так як систему легко переносити із місця на місце, та розгортка системи не потребує багато часу та зусиль. Розроблений модуль розгорнутий на одному з пристроїв системи.

Швидкодія модулю. Швидкодія модулю залежить від середовища розгортки. На даному апаратному забезпеченні модуль працює із задовільною швидкістю на невеликій кількості користувачів.

Модифікованість модулю забезпечується на високому рівні за рахунок використання гнучких технологій для побудови інтерфейсу та вдало спроектованої бази даних для їх підтримки.

КРИТЕРІЇ ОЦІНЮВАННЯ ЯКОСТІ РОЗРОБЛЕНОГО МОДУЛЮ

Кросплатформність модулю на високому рівні, так як модуль використовує технології що просто розгорнути на більшості існуючих операційних системах.

Універсальність застосування модулю. Модуль може застосовуватись у інших системах, що потребують частих модифікацій та кросплатформного інтерфейсу, так структура модулю побудована так, що можна швидко змінити інтерфейс та логіку роботи бази даних для нової системи.

ПОРІВНЯННЯ З АНАЛОГАМИ ЗА КРИТЕРІЯМИ



	Розроблена система	SCATT	MANTISX	MARKSMAN
Функціональність	7	8	6	5
Відмовостійкість	9	7	7	3
Мобільність	7	5	9	4
Швидкодія	7	10	10	10
Модифікованість	9	6	5	-
Кросплатформність	10	7	7	-
Універсальність застосування	9	5	5	-

Добре

10

9

8

7

6

5

4

3

2

1

Погано

18

РОЗВИТОК ПРОДУКТУ

- Додання користувацького доступу.
- Додання статистичного аналізу.
- Додання функції побудови вправ для стрільців.
- Додання планування занять до системи.
- Додання режиму для змагань.
- Додання функцій для відстеження стану системи.

ВИСНОВКИ

Розроблено комунікаційний модуль до системи динамічного визначення змін у стані об'єкту, де у ролі об'єкту виступає мішень. У ході роботи над системою було:

1. Проаналізовано альтернативні системи для вирішення задачі.
2. Розроблені вимоги до системи та модулів що розробляються.
3. Спроектовано та розроблено архітектуру системи та її окремих модулів.
4. Розроблено комунікаційний модуль до системи.
5. Розроблений модуль проаналізовано та порівняно із альтернативними системами за критеріями якості системи.

ДОСЯГНЕННЯ ТА ПУБЛІКАЦІЇ ПРОЕКТУ



Нагороди:

- Фіналіст та переможець фестивалю інноваційних розробок «Sikorsky Challenge 2018».
- Фіналіст фестивалю воєнних інноваційних розробок «Sikorsky Challenge 2019. Military».
- Підписано маніфест про наміри з фондом Міхалевича.
- Лауреат премії президентського фонду «Україна» Леоніда Кучми.

Публікації:

«Integer Norm for Difference Assessment of the Frame Elements Considering the White Balance», подана в редакцію наукового журналу «Управляющие системы и машины».

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Науковий керівник кафедри

_____ І.А. Дичка

“ ____ ” _____ 2018 р.

АВТОМАТИЗОВАНА СИСТЕМА ДИНАМІЧНОГО ВИЗНАЧЕННЯ
ЗМІН У СТАНІ ОБ'ЄКТА. КОМУНІКАТИВНИЙ МОДУЛЬ

Програма та методика тестування

ДП.045200-04-51

“ПОГОДЖЕНО”

Керівник проекту:

_____ Л.А. Люшенко

Нормоконтроль:

_____ М.В. Онай

Виконавець:

_____ Д.Д. Охримчук

ЗМІСТ

1. Об'єкт випробувань.....	3
2. Мета тестування	3
3. Методи тестування	3
4. Засоби та порядок тестування.....	4

1. ОБ'ЄКТ ВИПРОБУВАНЬ

Автоматизована система динамічного визначення змін у стані об'єкта. Комуникативний модуль, що є окремою частиною системи динамічного визначення змін у стані об'єкта, забезпечує зв'язок різних модулів між собою, надає інтерфейс, зберігає дані о роботі системи.

2. МЕТА ТЕСТУВАННЯ

У процесі тестування має бути перевірено наступне:

- можливість вести спостереження із власних пристроїв;
- ідентифікація та аналіз влучень у мішені;
- підтримка взаємодії із іншими модулями;
- відповідність інтерфейсу вимогам;
- можливість переглядати фото мішені;
- ідентифікація влучень;
- ознайомлення із результатами аналізу влучень;
- робота системи під навантаженням.

3. МЕТОДИ ТЕСТУВАННЯ

Тестування проводиться у декілька етапів. Спочатку тестуються окремі модулі системи, потім – їх взаємодія, і нарешті – робота всієї системи разом.

Мета даного методу тестування полягає у тому, що розробка окремих модулів також може супроводжуватись тестуванням, і за цим підходом поєднання окремих модулів, а потім – і всієї системи можливе лише коли складові частини ретельно перевірені.

Також проводиться тестування системи шляхом навантаження системи, що дозволяє перевірити границі можливостей системи у рамках існуючого обладнання.

Отже, під час тестування застосовуються наступні методи:

- функціональне тестування;
- тестування продуктивності та стабільності програмного забезпечення;
- тестування взаємодії різних модулів.

4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Під час розробки використовується мануальний спосіб тестування.

Порядок тестування системи має такий вигляд:

1. Тестування коректної роботи бібліотек шляхом перевірки правильної роботи набору тестових прикладів.
2. Перевірка роботи бази даних як сховища.
3. Перевірка роботи збережених процедур у базі даних.
4. Тестування взаємодії модулю бази даних із модулем аналізу та оптимізації.
5. Тестування інтерфейсу та його можливостей.
6. Тестування роботи інтерфейсу із базою даних.
7. Тестування загальної роботи системи.
8. Тестування системи під час роботи із декількома пристроями одночасно.
9. Тестування роботи системи під час роботи із більш ніж 10 користувачами.

Факультет прикладної математики
Кафедра програмного забезпечення комп'ютерних систем

“ЗАТВЕРДЖЕНО”

Науковий керівник кафедри

_____ І.А. Дичка

“ ____ ” _____ 2019 р.

**АВТОМАТИЗОВАНА СИСТЕМА ДИНАМІЧНОГО ВИЗНАЧЕННЯ
ЗМІН У СТАНІ ОБ'ЄКТА. КОМУНІКАТИВНИЙ МОДУЛЬ**

Керівництво користувача

ДП.045200-05-34

“ПОГОДЖЕНО”

Керівник проекту:

_____ Л.А. Люшенко

Нормоконтроль:

_____ М.В. Онай

Виконавець:

_____ Д.Д. Охримчук

ЗМІСТ

1. Доступ до інтерфейсу системи	3
2. Опис можливостей інтерфейсу системи	3
2.1. Перегляд інформації.....	3
2.2. Додання серій.....	4
2.3. Обрання серії.....	5
2.4. Обрання мішені	5
2.5. Оновлення сторінки	6

1. Доступ до інтерфейсу системи

Доступ до інтерфейсу систему можна отримати шляхом підключення будь-якого пристрою до локальної мережі, до якої також під'єднана система. Далі потрібно знати IP адресу серверу, що підключений до локальної системи, і ввести його у браузері. Далі користувач отримує інтерфейс із яким можна взаємодіяти із системою.

2. Опис можливостей інтерфейсу системи

Інтерфейс надає користувачу такі можливості:

- додавати нову серію. Ця функція створює нову групу для запису пострілів, що ідуть після часу її створення;
- обирати серію для перегляду. У залежності від обраної серії на екран буде виводитися постріли, що були записані у серії;
- обирати мішень для перегляду. Відповідно до обраного елемента із списку, обирається для перегляду необхідна користувачу мішень;
- перегляд довідки. При натисканні на кнопку «поміч» користувачу надається інструкція користувача;
- оновлення сторінки. Кнопка відповідає за оновлення структури інтерфейсу відповідно до обраних параметрів.

2.1. Перегляд інформації

У відповідних частинах інтерфейсу можна ознайомитись із інформацією, збереженою у системі щодо влучень (рис. 1, 2).



Рис. 1. Збережений стан мішені

Your hits			
Id	Target	Time	How far from center (mm)
1	0	2019-04-27 00:00:00	131.71
2	0	2019-04-27 01:00:00	87.13

Рис. 2. Влучення у серії

2.2. Додання серій

Додання серій відбувається шляхом натискання на відповідну кнопку (рис. 3).

Series number: series №1 ▼

target's number for show: target №0 ▼

☒ check current seria automaticaly

Refresh

Start new seria

Рис. 3. Додання серії

2.3. Обрання серії

Обрання серії для перегляду відбувається шляхом вибору серії із списку (рис. 4).

Series number: series №1 ▼

target's number for show: target №0 ▼

☒ check current seria automaticaly

Refresh

Start new seria

Рис. 4. Обрання серії

2.4. Обрання мішені

Обрання мішені для перегляду відбувається шляхом вибору мішені із списку (рис. 5).

Series number: series №1 ▼

target's number for show: target №0 ▼

☒ check current seria automaticaly

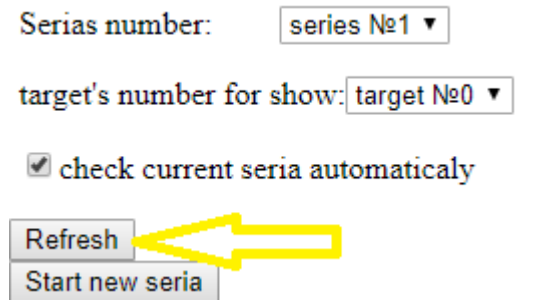
Refresh

Start new seria

Рис. 5. Обрання мішені

2.5. Оновлення сторінки

Оновлення сторінки відбувається шляхом натискання на кнопку (рис. 6).



Series number: series №1 ▼

target's number for show: target №0 ▼

☒ check current seria automaticaly

Refresh Start new seria

Рис. 5. Оновлення сторінки